Semi-Automation of a Rockbreaker System: Dynamic modeling and optimal collision-free trajectory planning

by

Louis-Francis Tremblay

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Applied Science (MASc) in Natural Resources Engineering

The Faculty of Graduate Studies
Laurentian University
Sudbury, Ontario, Canada

THESIS DEFENCE COMMITTEE/COMITÉ DE SOUTENANCE DE THÈSE
Laurentian Université/Université Laurentienne
Faculty of Graduate Studies/Faculté des études supérieures

Title of Thesis Titre de la thèse	Semi-Automation of a Rockbreaker System: Dynamic modeling and optimal collision free trajectory planning
Name of Candidate Nom du candidat	Tremblay, Louis-Francis
Degree Diplôme	Master of Science
Department/Program Département/Programme	Natural Resources Engineering
Date of Defence Date de la soutenance	June 28, 2018

APPROVED/APPROUVÉ

Thesis Examiners/Examinateurs de thèse:

Dr. Marc Arsenault
(Co-Supervisor/Co-directeur de thèse)

Dr. Meysar Zeinali
(Co-Supervisor/Co-directeur de thèse)

Dr. Brent Lievers
(Committee member/Membre du comité)

Dr. Roger Boudreau
(External Examiner/Examinateur externe)

Approved for the Faculty of Graduate Studies
Approuvé pour la Faculté des études supérieures
Dr. David Lesbarrères
Monsieur David Lesbarrères
Dean, Faculty of Graduate Studies
Doyen, Faculté des études supérieures

ACCESSIBILITY CLAUSE AND PERMISSION TO USE

Abstract

In light of technological advancements, the mining industry is seeing an increase in equipment automation. A hydraulic rockbreaker is a machine that would benefit from automation. The goal of this research is to develop some of the necessary algorithms to render a rockbreaker semi-autonomous. Semi-automation of such systems would allow for improved ease of use, increased productivity and efficiency of rock breaking operations, reduced maintenance costs while also removing the operator from harm's way. Several components are necessary to make semi-automation feasible, including a dynamic model as well as trajectory planning algorithms which generate collision-free trajectories to be used by a controller. The development of a complete dynamic model for such a system would allow for better control when using model-based controllers. However, such a model is difficult to develop in practice, has added complexity and may be computationally expensive. In this work, simplified dynamic models are developed and compared with respect to a complete dynamic model of the rockbreaker. One of the resulting simplified dynamic models, which only considers the inertial and gravitational effects of the rockbreaker's mechanical links as well as the gravitational effects of its hydraulic actuators, is shown to provide adequate representation of the system so as to be used in a model-based controller. The work also develops a set of offline trajectory planning algorithms that generate time-optimal trajectories which ensure smooth motions and hydraulic valve actuation while satisfying the system's flow rate constraints. With the addition of a collision avoidance strategy and collision detection algorithm, the generated trajectories within the system's work environment can be expected to be collision-free.

Keywords

Rockbreaker, Hydraulic Machinery, Automation, Dynamic Modeling, Trajectory Planning, Collision Avoidance, Collision Detection

Contents

List of Tables

List of Figures

Acknowledgments

The work presented herein is the result of the efforts of several people and organizations. The semi-automation of a rockbreaker system was made possible with Rock-Tech's desire for innovation within the mining sector with support from the Centre for Excellence in Mining Innovation (CEMI) through the Ultra-Deep Mining Network (UDMN) program.

To my supervisor and friend Dr. Marc Arsenault, my most sincere gratitude for the support provided throughout these past few years. Without his guidance, many insightful discussions and above all else his patience, I believe the completion of this work may not have occurred. To my co-supervisor Meysar Zeinali, many thanks for feedback provided on my work as well as insights on hydraulic systems.

To my friend and co-researcher Jasmin Lemieux, many thanks for allowing me the opportunity of pursuing this degree while offering advice and help in the development of numerous algorithms throughout this research project.

Thanks especially to my parents, François and Sylvie, my sisters Andréanne and Camille, whom offered me support throughout my years of postsecondary education and provided many laughs and wonderful memories.

To my girlfriend Jennifer, thank you for your unconditional love, support and patience throughout this chapter of my life.

Co-Authorship Statement

The following thesis has been written in a manuscript-based format. However, some additions and formatting changes were made to the original manuscripts to promote the coherence and clarity of the presented work. A list of the manuscripts, written during the extent of the research project presented herein and used to prepare this thesis, is provided here. The nature and scope of work of the co-authors for each manuscript is also provided.

Louis-Francis Tremblay, Marc Arsenault, and Meysar Zeinali. Simplification of the dynamic model of a hydraulic rockbreaker for the purpose of implementation in a model-based control scheme. *Transactions of the Canadian Society of Mechanical Engineerings (in press)*, 2018. [1]

Louis-Francis Tremblay was responsible for the kinematic and dynamic modeling of the rockbreaker system. Furthermore, Louis-Francis Tremblay ran the necessary simulations for the result section and was the main author on the paper. Marc Arsenault provided a MATLAB/SIMULINK model for the dynamic modeling of the rockbreaker system with the inclusion of the actuator dynamics. Furthermore, he revised the manuscript extensively during the writing process. Meysar Zeinali offered some feedback on the dynamic effects of the rockbreaker's mechanical links on the hydraulic system as well as valuable information regarding the expected behaviour of the hydraulic system. He provided some feedback during the revision process of the manuscript.

Louis-Francis Tremblay, Marc Arsenault, and Meysar Zeinali. Trajectory planning for a 4-DoF hydraulic rockbreaker based on flow limit constraints. *To be submitted for publication.*

Louis-Francis Tremblay was responsible for the development of the trajectory planning algorithm

as well as generating a sample trajectory provided in the paper. Louis-Francis Tremblay was the main author for this paper. Marc Arsenault provided some insights on the trajectory planning methodology throughout the development of the algorithm while revising the manuscript extensively during the writing process. Meysar Zeinali provided some feedback regarding trajectory planning algorithms as well as some revisions to the manuscript.

Marc Arsenault, Louis-Francis Tremblay, and Meysar Zeinali. Optimization of straight-line point-to-point trajectory durations based on flow rate scaling for a 4-DoF semi-automated hydraulic rock-breaker. *To be submitted for publication.*

Marc Arsenault was the primary author of the manuscript and developed the constraints described in Section 6.4.2. The work provides a description of Louis-Francis Tremblay's work regarding improvements to the trajectory planning algorithm seen in the aforementioned work. Louis-Francis Tremblay provided feedback during the writing process of the manuscript as well as generating trajectories utilizing the trajectory planning algorithms. Meysar Zeinali provided some feedback during the development of the algorithm.

Louis-Francis Tremblay, Marc Arsenault, and Meysar Zeinali. Semi-automation of a rockbreaker system: The development of a collision-free trajectory planning strategy. *To be submitted for publication.*

Louis-Francis Tremblay developed a collision-free trajectory planning strategy utilizing an enhanced form of the Gilbert-Johnson-Keerthi collision detection algorithm in conjunction with trajectory planning algorithms developed in the aforementioned works. Louis-Francis Tremblay was the primary author of the manuscript and generated the results for the paper. Marc Arsenault provided extensive feedback during the writing process of the algorithm, while Meysar Zeinali provided some insight on the subject of collision detection and collision avoidance during the writing process.

Nomenclature

Abbreviations

CA:	Cartesian straight-line motion approach
CAD:	Computer-aided design
DA:	Decoupled swing joint motion approach
DoF:	Degree-of-freedom
DKP:	Direct kinematic problem
FL:	Far left
FR:	Far right
GJK:	Gilbert-Johnson-Keerthi
GUI:	Graphical user interface
HA:	Hybrid approach
IKP:	Inverse kinematic problem
NL:	Near left
NR:	Near right
SQP:	Sequential-quadratic programming

Symbols

a:	Link length along longitudinal axis of link
$\mathbf{a}$:	Position vectors of points found within $\mathcal{A}$
$[\mathbf{a}_j]_i$:	Position vector of the $(j+1)^{\text{th}}$ reference frame relative to the i^{th} reference frame
A:	Actuator body pin connection
A:	Area of hydraulic actuator's piston upon which hydraulic fluid is exerting force upon
$\mathcal{A}$:	Set of points which make up a convex polytope
$\mathbf{A}$:	Matrix for the identification of $\mathbf{h}$ values
b:	Link offset
$\mathbf{b}$:	Constraint vector for the identification of $\mathbf{h}$ values
	Position vectors of points found within $\mathcal{B}$
B:	Actuator rod pin connection
$\mathcal{B}$:	Set of points which make up a convex polytope
c_i, c_{ij}, etc.:	Cosine of θ_i, Cosine of $\theta_i + \theta_j$, etc.
$[\bar{\mathbf{c}}]_1$:	Position of centre of mass relative to the base reference frame $X_1Y_1Z_1$
$\mathbf{c}$:	Position vectors of points found within $\mathcal{C}$
C:	Centre of mass
$\mathcal{C}$:	Minkowski sum of $\mathcal{A}$ and $-\mathcal{B}$
d:	Euclidean distance between two convex polytopes
$\mathbf{d}_{hj}$:	Position vector of a point j within $\mathcal{D}_h$

D_i: Placeholder variable i for complex equation

$\mathcal{D}$: Set of points defining collidable objects

$\mathbf{e}$:

Unit vector to isolate joint torque at R_2 for dynamic result simulation

Translational and angular velocities combined to obtain the Cartesian velocity vector as a function of $\dot{\sigma}$

$\mathbf{e}_g$: Unit vector parallel to gravitational field

f: Minimization function

g:

Gravitational acceleration

Optimization constraints for ramping segments used in the trajectory planning algorithm

$\mathbf{g}$: Gravitational effects vector

h: Coefficient of a polynomial

$\mathbf{h}$: Array of coefficients of a polynomial

$\mathbf{H}$: Upper triangular binary matrix used as lookup table for possible collision pairs

$\mathbf{i}$: Unit vector along X axis

$\bar{\mathbf{I}}$: Inertia matrix parallel to subsequent reference frame taken at the centre of mass

$\mathbf{j}$: Unit vector along Y axis

J_{i_j}: Element j of Jacobian matrix defined relative to i

$\mathbf{J}$: Jacobian matrix

m: Mass of element

$\mathbf{M}$: Generalized inertia matrix

$\mathbf{n}$: Unit vector along rockbreaker link

N: Number of discrete time-steps within a trajectory for ϕ trajectory verification

O: Origin of reference frame

$\mathbf{p}$: Cartesian position of point P relative to base reference frame $X_1Y_1Z_1$

$\mathbf{p}'$: Cartesian position of point P relative to reference frame $X_2Y_2Z_2$

P: Point located on breaker tip of rockbreaker

P': Point located along longitudinal axis of breaker's chisel

$\mathcal{P}$: Set of arc and line segments defining the boundary of admissible region

$\mathcal{P}_{1/2}$: Alternative paths which connect the initial and final rockbreaker configuration within $\mathcal{P}$

Q: Flow rate

$\hat{Q}$: Normalized flow rate

$\mathbf{Q}$: Array of flow rates

$\hat{\mathbf{Q}}$: Normalized flow rate vector

$\tilde{\mathbf{Q}}$: Optimal flow rate vector

r:

Constant scaling ratio

Discrete time increment

$\bar{r}$: Discrete time increment where the transition between constraining hydraulic component (*i.e.* hydraulic proportional valve or pump) occurs

R: Passive revolute joint

$\mathcal{R}_{a/g}$: Set of arc and line segments defining the annular region and grizzly perimeter

$\mathbf{R}^i_j$: Rotation matrix bringing reference frame j parallel to reference frame i

s: Tangent of the half-angle substitution variable

$s_{\mathcal{A}/\mathcal{B}/\mathcal{C}}$: Support function for set of points $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$

s_i, s_{ij}, etc.: Sine of θ_i, Sine of $\theta_i + \theta_j$, etc.

$[\mathbf{s}_j]_i$: Position of centre of mass j relative to reference frame i

t: Time variable

T: Total kinematic energy of the system

u:

Length of line segment defined by A_j and R_i

Minimal distance between O and all points found within a set of points (*e.g.* $\mathcal{C}$)

$\mathbf{u}$: Position vector of A_j relative to R_i in the i^{th} reference frame

U: Total potential energy of the system

v: Length of line segment defined by B_j and R_i

$\mathbf{v}$:

Coriolis and centrifugal effects vector

Search direction vector supplied to a support function

$\mathbf{w}$: Position vector of point W

W: Point within $\mathcal{C}$ located on the perimeter of the polytope defined by $\mathcal{C}$

$\mathcal{W}$: Set of points which make up a convex hull

$\mathbf{x}$: Cartesian pose of rockbreaker with respect to base reference frame $X_1Y_1Z_1$

x, y, z: Cartesian coordinates of the breaker tip (at point P) with respect to base reference frame $X_1Y_1Z_1$

X, Y, Z: Axes of orthogonal reference frames

X', Y', Z': Axes of orthogonal reference frames where X' is aligned with longitudinal axis of prismatic actuator

$\nearrow$: Increase in the value of a variable

Subscripts

a: Element where the maximum normalized flow rate is encountered during the ramp up motion

act: With reference to entire prismatic actuator

b:

With reference to an actuator barrel/body

Element where the maximum normalized flow rate is encountered during the ramp down motion

β: Jacobian matrix establishing relationship between $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$

blend: Total duration of blending segment at $\bar{r}$

blind: Area of blind side of piston

c: Constant velocity segment

f/F: Final state of an element

i: Rockbreaker link index number

I: Initial state of an element

$I/II/III$: Simplified dynamic model version number

j: Rockbreaker actuator index number

$l/link$: With reference to a link

lim: Limit value of variable'

p:

With reference to the hydraulic pump

With reference to a piston assembly

Jacobian matrix establishing relationship between $\mathbf{Q}$ and $\mathbf{Q}_v$

θ: Jacobian matrix establishing relationship between $\mathbf{x}$ and $\boldsymbol{\theta}$

q:

Jacobian matrix resulting from $\mathbf{J}_p\mathbf{J}_v$

Quarter points of variable's total value (*i.e.* σ =0.25, 0.5, 0.75)

ρ: Jacobian matrix establishing relationship between $\boldsymbol{\beta}$ and $\boldsymbol{\rho}$

rod: Area of rod side of piston

v: Flow rate through proportional valves

$\downarrow$: Ramp down segment element

$\uparrow$: Ramp up segment element

Superscript

m: Dimension of coordinate space

*: Alternative Jacobian matrix as $\mathbf{J}_\beta$ does not have an inverse

$\breve{}$: Scaled variable

Greek letters

α: Link twist angle

β: Angle obtained by applying the cosine law to the triangle formed by nodes A_j, B_j and R_i

$\boldsymbol{\beta}$: Array of all β_j angles

γ: Angle between axis X_i and X'_j

δ: Sign relationship between θ_i and β_j (dependent on actuator mounting configuration)

Δ: Absolute difference

ϵ: Constant angular offset for to compute θ

ε: Relative joint torque error

η: 3^{rd}-order polynomial blend function of scaling ratio

θ: Passive revolute joint angle

$\boldsymbol{\theta}$: Array of joint angles θ

λ: Scaling ratio function over trajectory

$\boldsymbol{\lambda}$: Design vector of optimization function

ξ: Percentage factor for ϕ_q adjustment

ρ: Prismatic actuator length

$\boldsymbol{\rho}$: Array of prismatic actuator lengths ρ

σ: Monotonically increasing parameter where $\sigma \in [0, 1]$

τ: Joint torque

$\boldsymbol{\tau}$: Array of joint torques

ϕ: Orientation angle of breaker tip relative to X_2 axis

φ: Angle offset between breaker's longitudinal axis and X_5 axis

ψ: Angle offset between X_2 and X_5

$\boldsymbol{\omega}$: Absolute angular velocity relative to base reference frame $X_1Y_1Z_1$

Chapter 1

Introduction

Throughout history, mining has enabled humans to obtain materials that were once unattainable in order to further technological advancements [2]. To this day, mining remains an integral part of modern society. As demands for mineral resources continue to rise, mineral extraction operations must increase their throughput in order to satisfy these demands. This can be achieved through mechanization and automation of mining processes and equipment. One piece of equipment that is of interest for automation is the hydraulic rockbreaker (as seen in Figure 1.1). This machine can be perceived as a four-degree-of-freedom (4-DoF) hydraulically-driven robot arm normally anchored to a stationary pedestal, which is used extensively in the mining and construction industry as a means to fragment large pieces of material. The tool used by the rockbreaker to fragment rock is commonly referred to as the *breaker*. During the mining process, pieces of material are transported throughout the mine by various means. However, large pieces of material that cannot easily be transported or broken down into smaller fragments using a conventional jaw crusher must be dealt with using a hydraulic rockbreaker. This entails placing the material on a metallic grid, known as a grizzly, which allows the smaller pieces to fall through down a chute to a lower level in the mine (known as an ore pass) or to the jaw crusher, whereas the bigger fragments remain on the grizzly. The large pieces of material are then broken using percussive force supplied through the rockbreaker's chisel.

The motivations behind automating such a system are numerous. These include an increase in productivity, operation efficiency, workplace safety [2–27] as well as a lower operating cost of the machine [11]. Furthermore, automation of such a system is not uncommon in this day and age,

Figure 1.1: Typical rockbreaker installation in underground mining operations. Image taken (with permission) from http://www.rock-tech.net/products/stationary-rockbreaker-systems/xd-series (accessed February 16$^{\text{th}}$, 2018)

where various industries such as logging, agriculture, construction and mining have benefited from automation of similar hydraulic machines [3–10]. Current automation technology that is used on rockbreaker systems consist of teleoperation [17], where a qualified operator can operate the machine from a distance (*e.g.* from above ground or from an office building located hundreds of kilometers away from the actual machine [11]). However, this technology has some drawbacks, such as the introduction of latency issues as well as a reduction of spatial awareness for the operator [28]. Furthermore, teleoperation still necessitates a trained operator, which is harder to come by due to the steep learning curve associated with operating such machines [3, 29]. To mitigate the shortcomings of teleoperation, a supervisory control system can be implemented [4, 17], allowing for a more intuitive control of the machine (*e.g.* such as by allowing desired Cartesian motions as inputs from the operator). Such systems are more practical and can be expanded upon in order to allow for further ease of use for operators [4, 11–13, 16, 17, 30]. Seeing as there has been significant research regarding

automation of hydraulic excavators [16, 25] (which share many similarities with rockbreakers), elements from these findings can be adapted in order to further increase the level of automation for a rockbreaker, similar to what was done in [11].

The objective of this research, more precisely, is to develop a semi-autonomous rockbreaker system such that it may be retrofitted on a commercially available rockbreaker. The term *semi-autonomous* is synonymous to the automation of broad motions of the rockbreaker (within a static environment) between rock breaker operations. The operator would specify a desired end position for the rockbreaker's breaker tip through a graphical user interface (GUI), after which the system would generate a time-based, collision-free trajectory. This trajectory should ensure smooth and continuous hydraulic valve actuation (which in turn controls the hydraulic actuators) to reduce wear and tear within the hydraulic system. Furthermore, it should ensure the breaker tip remains with a horizontal work plane located above the grizzly (where no collisions between the rockbreaker and material on the grizzly are likely to occur) for the entirety of the motion. The resulting trajectory is then implemented by a controller where, upon the completion of the motion, the operator takes over the controls via conventional teleoperation to break the oversized piece of ore found on the grizzly. In order to achieve such a system, several components must be developed such as: dynamic modeling of the system, trajectory planning algorithms, collision avoidance algorithms, control algorithms, as well as the installation of hardware (*e.g.* control module, sensors, etc.). In this work, the development of an adequate dynamic model for a rockbreaker and the development of algorithms producing collision-free time-optimized trajectories are presented. It should be noted that the resulting semi-automated system can be considered as a stepping stone towards the full automation of a rockbreaker system which, in itself, involves numerous additional challenges such as the execution of non-repetitive tasks within dynamic environments [2, 20] as well as the development of a robust vision based system to identify the boulders that need to be fragmented [31,32].

The remainder of this thesis is organized as follows. Chapter 2 provides a detailed literature review of the dynamic modeling of hydraulic machinery such as rockbreakers and hydraulic excavators as well as the trajectory planning, collision detection and collision avoidance strategies used for the automation of such machines. Chapter 3 pertains to the general description of the geometry of a rockbreaker as well as a general kinematic analysis of the system. Chapter 4 gives an in-depth

description of the dynamic modeling of the rockbreaker, followed by the development of a preliminary trajectory planning algorithm for the rockbreaker based on a simple flow rate scaling approach in Chapter 5. A time-optimal trajectory planning algorithm based on a more sophisticated flow rate scaling approach is then introduced in Chapter 6, with the collision avoidance strategy for this system being discussed in Chapter 7. This is followed by a general conclusion in Chapter 8.

Chapter 2

Literature review

Research in the automation of hydraulic rockbreakers and similar hydraulic machines has been extensive over the years. A brief summary of the work done and reasoning behind the development of dynamic models, trajectory planning algorithms as well as collision avoidance strategies for such machines is given in the subsequent sections.

2.1 Dynamic modeling

In order to achieve suitable control of hydraulic rockbreakers and similar machines (*e.g.* hydraulic excavators), nonlinear control algorithms are typically designed with compensators and/or are based on the system's behaviour during operation as described by its dynamic model. Due to the presence of complex phenomena such as joint friction, parametric uncertainties and the non-linearity of the hydraulic and mechanical subsystems, the modeling and design of model-based controllers for such systems is a difficult and challenging task. Parameter uncertainties may lead to poor system response (*e.g.* higher trajectory tracking error) during the system's motion. However, such negative effects can be reduced through the use of a robust controller [33] or an adaptive controller [34, 35], which utilize the system's dynamic model in a parameterized form to estimate the uncertain parameters of the dynamic model. Therein lies the importance of developing an accurate yet not overly complex dynamic model such that it may be parameterized.

There have been numerous papers describing the dynamic modeling of hydraulic excavators, shovels

and backhoes in the past, based on following the Newton-Euler conventions [36–38], Kane's equations [39] or the Lagrange-Euler convention [40]. The dynamic model of a hydraulic excavator was obtained using the Newton-Euler method under the assumption that the excavator does not interact with its environment while the centre of mass for each link is taken to be in-line with each link's corresponding axis [36]. Such assumptions tend to oversimplify the system. To mitigate this, a more complete dynamic model of an excavator, where the centres of mass of each link are not assumed to be located along the links' longitudinal axes and resistive forces encountered during digging operations are taken into account, was developed [37]. The resulting model is said to be adequate for direct application on a real excavator system. The dynamic modeling of a hydraulic shovel (similar to an excavator, where the bucket faces outward rather than inward) in [38] provides a similar model to that of [37] but it also considers the dynamic forces resulting from the excavation process. It may be noted that the resistive forces are at their greatest during the initial moments of the excavation process. For the dynamic modeling using Kane's equation, [39] provides a model which considers resistive forces resulting from the excavation as well as the effects of soil deformation under the excavator during this process (*e.g.* excavator settling in soft soil). It was noted that the dynamic model's accuracy was heavily dependent on the parameter values of the system due to the geometric complexity of the system. Parameter values can be obtained through various means, using documentation and physical measurements of a real system to obtain link length and mass information, while inertial properties are obtained using CAD models, link geometry approximations or experimental parameter identification techniques [39]. Lastly, [40] describes the modeling of a hydraulic backhoe using the Lagrange-Euler approach, where the system model considers the excavation forces but omits the frictional forces in the system (*e.g.* seal friction within the hydraulic pistons). Dynamic simulations for a backhoe based on the resulting model as well as the model provided by [37] were conducted with both yielding similar results. It was noted that one of the joint torques computed by the Lagrange-Euler method varied by nearly 9% from the model provided in [37]. This discrepancy was attributed to the variance in geometry between an excavator and a backhoe.

Although the modeling process varied between [36–40], these models all represented the hydraulic machines as a planar three-degree-of-freedom (3-DoF) manipulator (where the swing joint of the machines is not considered). They also neglected the effects of the hydraulic actuators on the

dynamic model. In order to properly model the dynamics of a hydraulic rockbreaker, the effects of its hydraulic actuators should be taken into account. Furthermore, the rockbreaker should also be considered as a 4-DoF manipulator (rather than a 3-DoF manipulator) given that it is stationary and must rotate around its swing joint to position itself over a piece of material (as compared to an excavator/shovel/backhoe that is mobile and can easily be moved and positioned in an optimal digging configuration). As the semi-automation of the rockbreaker does not involve interactions with its environment, there is no need for the modeling of interaction forces between the breaker tip and the piece of material that needs to be broken.

2.2 Trajectory planning

In order to automate hydraulic systems such as rockbreakers and excavators, trajectory planning algorithms must be developed [16, 25]. These trajectory planning algorithms are used to create motion paths (*i.e.* time histories of actuator displacements, velocities and accelerations) that can be supplied to a controller. In many cases, the main objective of the trajectory planning algorithm is to minimize the trajectory duration. However, algorithms may also seek other goals such as the minimization of required joint forces or torques (*e.g.* [41, 42]). One challenge in developing suitable trajectory planning algorithms for such machines is the nonlinear nature of their actuation schemes [43]. In addition, when developing a trajectory planning algorithm for a hydraulic system, the system's constraints should be considered. This includes, for instances, constraints such as the geometric limitations of the system to reach a desired configuration (position and orientation of the end-effector) as well as the system's maximum allowable joint velocity (which is configuration dependent). Such an approach is used in [44] where the trajectory planning algorithm is based on a virtual motion camouflage approach. Furthermore, limits associated to hydraulic actuator flow rates should also be considered. In [19] as well as in the initial trajectory planning algorithm developed for this work (presented in Chapter 5), these limits are satisfied by proportionally scaling entire trajectories based on an identified critical point where flow rate demand is greatest. Several trajectory planning strategies for rockbreakers or other similar systems have been developed throughout the years with a summary review provided in [27]. These include the use of hard-coded predefined trajectories [11], rule-based motion planning based on simple commands that are sent to low-level

joint-based controllers according to a discrete set of states for each joint [14], the use of fuzzy logic representation of expert human knowledge to generate trajectories for different parts of an excavator's digging operation [20], the use of neural networks to generate and optimize trajectories for repetitive motions [43], the use of fifth-order polynomials to generate trajectories in the system's joint space [45] and, more commonly, the generation of trajectories using point-to-point motions in task (or Cartesian) space [13, 15, 17, 19, 37]. It is noted that task (or Cartesian) space trajectories can easily be converted into joint space trajectories for implementation in a control algorithm.

There are limitations associated to some of the aforementioned trajectory planning strategies. For instance, the technique presented in [45] utilizes a quintic polynomial (and its derivatives) to generate displacement, velocity and acceleration trajectories in joint space for each revolute joint of an excavator. This results in trajectories that do not take full advantage of the joints' maximum allowable velocities (or, equivalently, the available flow rates of the system) throughout the majority of the trajectory. This increases the total duration of the motion, making this approach less than optimal. Furthermore, due to the non-convex nature of a typical rockbreaker's workspace, using Cartesian-based straight-line trajectories such as the ones presented in [15, 17, 19, 37] may not always be feasible. In instances where Cartesian straight-line motions are infeasible due to not being located entirely inside the rockbreaker's workspace, a robust system should generate an alternative trajectory (rather than simply sounding an alarm for an operator to take over the operation of the machine as is the case in [19]). A common thread among much of the existing literature dealing with the trajectory planning of rockbreakers or excavators is an omission of the necessary details for the reproduction and utilization of the proposed methodologies, *e.g.* [13, 46].

Given that rockbreakers and other similar machines having some level of automation can be considered as robots, the extensive body of knowledge on robot trajectory planning (see, for instance, [47, 48]) can be utilized as a basis for the development of suitable trajectory planning algorithms. However, one must remain aware that conventional robot trajectory planning algorithms are tailored for systems equipped with electric servomotors (which are the most common motors utilized in robotic applications [33]) rather than a hydraulic actuation scheme such as the one present on the rockbreaker system. Both actuation systems have advantages and limitations which must be considered. The use of electric servomotors in terms of trajectory planning eases the interfacing and

control of the actuator, which can act as pure torque sources [33,35], although this latter is usually due to a reduction of the servomotor's motion through a gearbox. Having these acting as pure torque sources allows for generated trajectories to incorporate instantaneous joint accelerations [35] (although practically speaking, the joint accelerations are more akin to being quasi-instantaneous). When using a hydraulic actuation scheme however, the aforementioned assumptions are not always valid. Hydraulic actuation is favored for systems under heavy load, as they can provide greater power-to-weight ratios compared to electric servomotors [33]. However, these systems are inherently slow compared to conventional robotic manipulators due to their use of prismatic actuators (indirectly actuating the system's revolute joints), where their motion behaviour is highly dependent on the dynamic effects within the hydraulic system. Furthermore, compressibility of the hydraulic fluid within the system as well as friction between the actuators' seals make accurate control of the force applied by the actuators difficult to achieve [35] while also making instantaneous accelerations near impossible to achieve. Even if greater control of applied force was possible for hydraulic systems, instantaneous accelerations are to be avoided, since this would result in the rapid closure of hydraulic control valves, which are under high pressures. This leads to the creation of shock waves throughout the hydraulic systems, whose pressure levels can be four times greater than the designed steady-state pressure of the system, potentially causing serious damage to several components within the hydraulic system [49]. Therefore, if one is to apply conventional trajectory planning algorithms to hydraulic systems, and more specifically a rockbreaker, one should ensure trajectories generated ensure smooth and continuous hydraulic valve actuation, non-instantaneous joint accelerations while taking into account configuration dependent joint velocity limits.

2.3 Collision avoidance

As previously stated, a key component in order to develop a semi-automated hydraulic rockbreaker consists of a robust trajectory planning algorithm. Examples of such algorithms for rockbreaker systems as well as similar hydraulic systems can be seen in [13,19,20,37] as well as in the algorithms presented in this work. However, these algorithms fail to account for the possibility of collisions between the rockbreaker and its surroundings. Therefore, the inclusion of a collision avoidance strategy in conjunction with the trajectory planning algorithms is necessary to ensure safe, efficient

and optimal control of the rockbreaker.

When developing robotic manipulators, the possibility of collisions must be factored into the design process. Some collisions may be deemed desirable, *e.g.* a hydraulic excavator's bucket digging soil, or undesirable, *e.g.* a hydraulic excavator hitting a dump truck during a loading operation. For the case of a semi-automated rockbreaker, unwanted collisions can have devastating results due to the powerful nature (*e.g.* high power-to-weight ratio, relative tool tip velocity, etc.) of these machines [21,23,50]. In particular, excessive damage to the system or its environment (*e.g.* reduced structural integrity of surrounding walls) as well as serious bodily injuries to anyone found within the rockbreaker's workspace are possible, making the implementation of a collision avoidance strategy imperative for such a machine.

Although research in the automation of rockbreakers is still in its infancy, research in the automation of a similar systems (*i.e.* hydraulic excavators) is much more common with collision avoidance strategies having already been implemented, *e.g.* [2, 11, 14, 17, 21–24]. However, real-time collision avoidance strategies, where collision avoidance and trajectory planning occur simultaneously, have been used in most of these cases [11, 14, 17, 23, 24]. In the case of the rockbreaker system presented in this work, trajectories are precomputed via an offline trajectory planning algorithm to ensure smooth and continuous valve actuation. Real-time collision-avoidance strategies are not compatible with this approach. However, a collision detection algorithm may be integrated with the offline trajectory planning algorithms to generate collision-free trajectories.

There are two main components to a collision avoidance algorithm: the collision detection and the identification of alternative paths of motion when collisions are detected. Collision detection can be achieved through either physical detection of collisions using sensors (*e.g.* [50]) or through the use of collision detection algorithms. Such algorithms require the positional awareness of a machine's links through the use of sensors [11, 14, 17, 21–24] along with knowledge of the geometrical properties of the machine and any surrounding obstacles. Since the trajectories are known to be generated using offline trajectory planners, the use of a collision detection algorithm is preferable in this instance. Several collision detection algorithms exist to detect collisions between pairs of objects such as the sweep and prune algorithm using axis-aligned bounding boxes [17,51–53], the separate axis theorem

using object oriented bounding boxes [51,54], the Lin-Canny method and its variations [52,55,56] and the Gilbert-Johnson-Keerthi (GJK) distance algorithm and its variations [51,53,57–60]. Each collision detection algorithm has its benefits and drawbacks. The sweep and prune algorithm utilizing the axis-aligned bounding boxes as well as the separate axis theorem using object oriented bounding boxes are generally used in a broad phase collision detection (*i.e.* identifying the possibility of collisions between two objects) and where the bounding boxes are very conservative estimates of each collidable object. If a collision is found to occur between two objects, the bounding boxes can further be reduced into smaller bounding boxes and checked once more for collisions. This can be repeated until the resulting bounding boxes yield a close approximation to the modeled objects. However, there is an added complexity in programming such algorithms. The Lin-Canny algorithm and its variances as well as GJK algorithm are defined as narrow phase[1] collision detection. These algorithms may be computationally demanding compared to the broad phase algorithms for environments with numerous collidable objects, however, these offer a better representation of the collidable objects and therefore, a better collision detection fidelity. The Lin-Canny algorithm and its variants, for their part, have some drawbacks. In particular, the algorithm may not terminate when presented with a case where there is penetration between two collidable objects [52]. As for the enhanced GJK algorithm, its implementation is simple, it is robust and with coherence, can be computationally faster than the Lin-Canny algorithm [57] provided by the library developed in [61]. Furthermore, this algorithm has also been implemented in robotics, graphics and physics engines [62], including the *Bullet* physics engine [60], which has been used extensively in the gaming and movie industries [63]. Since the number of collidable objects within the rockbreaker's environment is expected to be minimal and collision detection accuracy is crucial, the enhanced form of the GJK algorithm is an excellent candidate to be used in the development of a collision avoidance strategy for the rockbreaker system presented in this work.

[1]The term narrow phase collision detection is used to define an algorithm which accurately detects a collision between a pair of objects of complex geometry. In contrast, broad phase collision detection seeks to identify potential collisions and collision pairs between objects using conservative estimates of the objects' dimensions (*i.e.* representing an object as a sphere). These algorithms can be used in conjunction to efficiently detect collisions, where the broad phase collision detection algorithm identifies collisions pairs which are then verified by the narrow phase collision detection algorithm.

Chapter 3

Geometrical description and kinematic analysis of a rockbreaker

The kinematics of a system give insight on the motion of the components found within a system, describing the position, velocity and acceleration of these components within a specific coordinate system. The modeling of the rockbreaker's kinematics is a crucial element in the creation of a semi-autonomous rockbreaker system, playing a vital role in the development of dynamic models, trajectory planning algorithms and in object modeling necessary for the implementation of collision avoidance algorithms. A geometrical description and general overview of the kinematic analysis for a rockbreaker system is provided in this chapter. This includes solutions to its direct and inverse kinematic problems as well as insights on input–output velocity relationships, workspace and flow rate analyses.

3.1 Geometrical description

A CAD model of a typical rockbreaker installation is shown in Figure 3.1 where the base (typically encased in a concrete foundation), swing post, inner boom, outer boom and breaker may be seen to constitute the links of a 4-DoF serial-type robot with revolute joints. A schematic representation of the rockbreaker is provided in Figure 3.2 where the passive revolute joints have been labeled as R_i $(i = 1, 2, 3, 4)$ with corresponding angles, measured between consecutive links, denoted as θ_i. The

link lengths a_i correspond to the perpendicular distances between the axes of consecutive revolute joints with the exception of a_4, which is measured from the axis of R_4 to a point P located at the tip of the breaker's chisel. Each of the rockbreaker's revolute joints is driven by a series of hydraulically driven prismatic actuators of length ρ_j $(j = 0, 1, \ldots, 4)$, measured as the distances between the actuator body (A_j) and rod (B_j) pin connections as seen in Figure 3.2. This actuation scheme allows for the efficient generation of large joint torques at the cost of nonlinear relationships between the θ_i and ρ_j. It should be noted that, for the actuation of R_1, two actuators (ρ_0 and ρ_1) work in parallel (see Figure 3.2(a)) in order to generate a symmetric torque v. angular displacement curve. The hydraulic actuators and revolute joints are mechanically limited such that $\rho_{j_{\min}} \leq \rho_j \leq \rho_{j_{\max}}$ and $\theta_{i_{\min}} \leq \theta_i \leq \theta_{i_{\max}}$.

The reference frame assignment for the rockbreaker is based on the Denavit-Hartenberg convention described in [64], where the i^{th} reference frame is fixed to the $(i-1)^{\text{th}}$ link with its origin located at R_i (with the exception of the $X_5Y_5Z_5$ reference frame, which is located at P), where the Z_i axis is directed along the axis of R_i and where the X_i axis is aligned such that it is perpendicular to the Z_i and Z_{i+1} axes. For example, the reference frame $X_1Y_1Z_1$ is fixed and set as the base reference frame, with its origin located at point O and is defined such that the Z_1 axis is directed along the axis of R_1 while the X_1 axis is oriented so that the rockbreaker would be located in the X_1Z_1 plane when $\theta_1 = 0$. Meanwhile, the second reference frame $X_2Y_2Z_2$ is attached to the swing post with the Z_2 axis coinciding with the axis of R_2 with Y_2 directed vertically upward. Subsequent reference frames follow the previously mentioned convention and can be seen in Figure 3.2. The inputs to the rockbreaker system are considered to be the hydraulic actuator lengths ρ_j whereas its output pose is represented as the combination of the position $\mathbf{p}$ of the chisel tip P measured from O as well as the orientation ϕ of the breaker. The latter is measured from the X_2 axis to a line parallel to the longitudinal axis of the chisel (*i.e.* the line joining nodes P and P' in Figure 3.2(b)).

The hydraulic actuators of lengths ρ_0 and ρ_1 that are used to drive the rockbreaker's swing joint (*i.e.* R_1) are controlled by a common proportional valve whose variable flow rate is Q_1. The remaining hydraulic actuators, for their part, are controlled by their own individual proportional valves with variable flow rates Q_i $(i = 2, 3, 4)$. The rockbreaker's velocity is constrained by the maximum flow rates $Q_{i_{\max}}$ $(i = 1, 2, 3, 4)$ through each of these proportional valves as well as the maximum

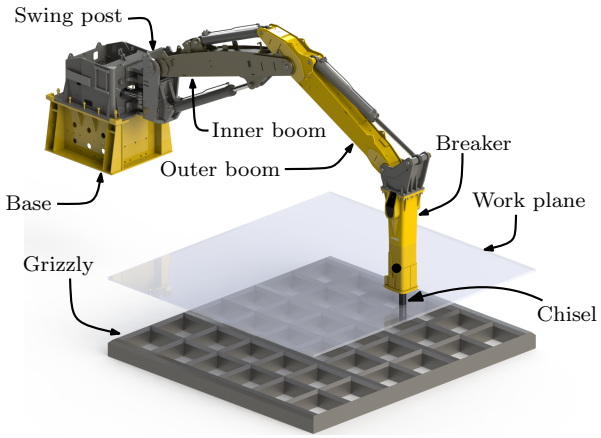


Figure 3.1: Typical rockbreaker installation in underground mining operations.

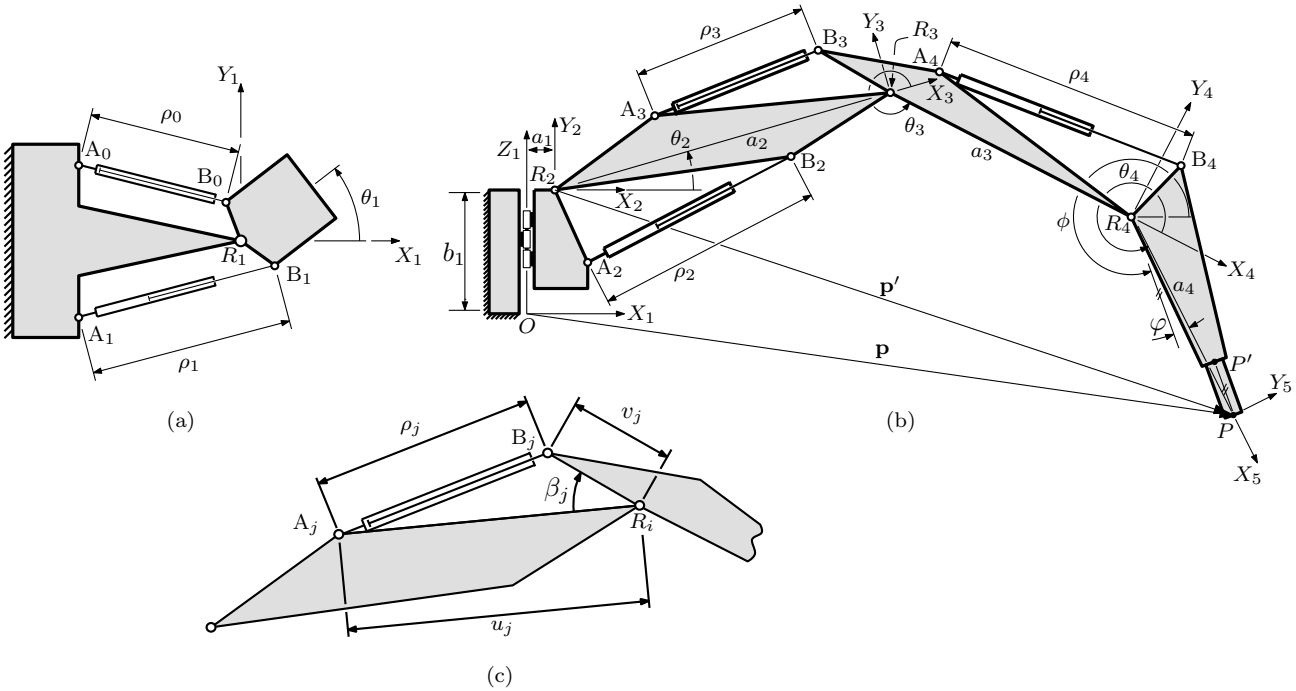


Figure 3.2: Schematic diagram of the rockbreaker: a) Top view of base and swing post, b) side view of rockbreaker and c) detailed side view of a typical hydraulic cylinder-actuated joint (adapted from [1]).

throughput of the hydraulic pump $Q_{p_{\max}}$. It should be mentioned that flow rate limits play a crucial role in planning trajectories for the rockbreaker system, where these limits must be respected throughout any given trajectory.

3.2 Direct kinematic problem

The rockbreaker's direct kinematic problem (DKP) is defined here as the computation of the pose of its breaker (*i.e.* $\mathbf{p}$ and ϕ) for given actuator lengths ρ_j. Referring to the schematic representation of a typical joint shown in Figure 3.2(c), the angle β_j may be obtained by applying the cosine law to the triangle formed by nodes A_j, B_j and R_i, *i.e.*

$$\beta_j = \cos^{-1}\left(\frac{u_j^2 + v_j^2 - \rho_j^2}{2u_j v_j}\right) \tag{3.1}$$

where u_j defines the distance between R_i and A_j while v_j defines the distance between R_i and B_j ($i = 1, 2, 3, 4$ and $j = 0, 1, \ldots, 4$). The i^{th} revolute joint angle θ_i may then be obtained as

$$\theta_i = \delta_j \beta_j + \epsilon_j \tag{3.2}$$

where ϵ_j is a constant angular offset based on the system's geometry whereas δ_j is the $(j+1)^{\text{th}}$ element of $\boldsymbol{\delta} = [-1, 1, 1, -1, -1]^T$, which is determined by the mounting location of the j^{th} prismatic actuator with respect to the i^{th} revolute joint. It is worthwhile noting here that only one of ρ_0 and ρ_1 is necessary to compute θ_1 due to the kinematic redundancy.

With θ_i known, the position of the $(i+1)^{\text{th}}$ frame relative to the i^{th} frame can be obtained as:

$$[\mathbf{a}_i]_i = [a_i \cos\theta_i, a_i \sin\theta_i, b_i]^T \tag{3.3}$$

where $[\cdot]_i$ represents a vector expressed in the i^{th} reference frame and where b_i represents the link offset, where b_1 is shown in Figure 3.2(b) while $b_2 = b_3 = b_4 = 0$. Meanwhile, the rotation matrix which brings the i^{th} frame parallel to the $(i+1)^{\text{th}}$ frame is computed as:

$$\mathbf{R}_{i+1}^{i} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix} \tag{3.4}$$

where α_i represents the link twist angle, where $\alpha_1 = 90°$ while $\alpha_2 = \alpha_3 = \alpha_4 = 0$. With Eqs. (3.3) and (3.4), the position of the breaker tip P may obtained as

$$\mathbf{p} = \sum_{i=1}^{4} [\mathbf{a}_i]_i = [\mathbf{a}_1]_1 + \mathbf{R}_2^1[\mathbf{a}_2]_2 + \mathbf{R}_2^1\mathbf{R}_3^2[\mathbf{a}_3]_3 + \mathbf{R}_2^1\mathbf{R}_3^2\mathbf{R}_4^3[\mathbf{a}_4]_4 \tag{3.5}$$

which simplifies to

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} c_1(a_1 + a_2c_2 + a_3c_{23} + a_4c_{234}) \\ s_1(a_1 + a_2c_2 + a_3c_{23} + a_4c_{234}) \\ b_1 + a_2s_2 + a_3s_{23} + a_4s_{234} \end{bmatrix} \tag{3.6}$$

where $c_i = \cos\theta_i$, $s_i = \sin\theta_i$, $c_{ij} = \cos(\theta_i + \theta_j)$, $s_{ij} = \sin(\theta_i + \theta_j)$, etc.. The breaker's orientation, for its part, is found as

$$\phi = \theta_2 + \theta_3 + \theta_4 - \varphi \tag{3.7}$$

In the above, φ is a constant angular offset illustrated in Figure 3.2(b). The overall pose of the rockbreaker in Cartesian space can be defined as $\mathbf{x} = [\mathbf{p}^T, \phi]^T$.

3.3 Inverse kinematic problem

The rockbreaker's inverse kinematic problem (IKP) is defined here as the computation of the actuator lengths ρ_j given the pose of the breaker ($\mathbf{p}$ and ϕ). The inverse kinematic problem (IKP) is useful from a trajectory planning point of view as it allows the conversion from Cartesian space to joint space of specified trajectories at the displacement level. This also allows for the verification of whether breaker poses are located within the rockbreaker's workspace. Given the position $\mathbf{p}$ of the chisel tip, the first revolute joint angle may be found as

$$\theta_1 = \text{atan2}(y, x) \tag{3.8}$$

In order to compute the remaining revolute joint angles, the position of the chisel tip measured from the origin of the $X_2Y_2Z_2$ frame and expressed in this frame is first computed as

$$\mathbf{p}' = (\mathbf{R}_2^1)^T(\mathbf{p} - [\mathbf{a}_1]_1) \tag{3.9}$$

From the perspective of the $X_2Y_2Z_2$ frame, the rockbreaker is simply a planar serial robot arm whose IKP is readily available in existing literature (*e.g.* [65]) such that only a geometrical description of

the approach is provided here. The combination of $\mathbf{p}'$ and ϕ allows for the determination of the position of R_4 in the $X_2Y_2Z_2$ frame. Afterwards, joint angles θ_2 and θ_3 may be found by intersecting two circles of radii a_2 and a_3 centered at nodes R_2 and R_4, respectively. Although this generally yields up to two distinct solutions (assuming the rockbreaker is operating within its workspace), only the "elbow-up" solution is feasible due to the system's construction. Once θ_2 and θ_3 are known, θ_4 may be found rather easily from Eq. (3.7). Afterwards, the hydraulic actuator lengths are obtained by solving Eq. (3.1) for ρ_j where the β_j are obtained from Eq. (3.2).

3.4 Velocity analysis

The trajectory planning methodologies that are presented in the later chapters of this work require conversions from the rockbreaker's translational and angular velocities in Cartesian space (*i.e.* $\dot{\mathbf{x}} = [\dot{\mathbf{p}}^T, \dot{\phi}]^T$) to corresponding hydraulic actuator velocities (*i.e.* $\dot{\boldsymbol{\rho}} = [\dot{\rho}_0, \dot{\rho}_1, \dot{\rho}_2, \dot{\rho}_3, \dot{\rho}_4]^T$). However, for completeness, the conversion of $\dot{\boldsymbol{\rho}}$ to $\dot{\mathbf{x}}$ will also be introduced.

Defining $\dot{\boldsymbol{\beta}} = [\dot{\beta}_0, \dot{\beta}_1, \dot{\beta}_2, \dot{\beta}_3, \dot{\beta}_4]^T$, one has $\dot{\boldsymbol{\beta}} = \mathbf{J}_\rho \dot{\boldsymbol{\rho}}$ where, referring to Eq. (3.1), $\mathbf{J}_\rho$ is a diagonal matrix whose $(j+1)^{\text{th}}$ element is given by

$$J_{\rho_{j+1}} = \frac{\partial \beta_j}{\partial \rho_j} = \frac{\rho_j}{u_j v_j \sin \beta_j} \tag{3.10}$$

The vector of revolute joint velocities, *i.e.* $\dot{\boldsymbol{\theta}} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4]^T$, may then be obtained as $\dot{\boldsymbol{\theta}} = \mathbf{J}_\beta \dot{\boldsymbol{\beta}}$ where

$$\mathbf{J}_\beta = \begin{bmatrix} J_{\beta_0}/2 & J_{\beta_1}/2 & 0 & 0 & 0 \\ 0 & 0 & J_{\beta_2} & 0 & 0 \\ 0 & 0 & 0 & J_{\beta_3} & 0 \\ 0 & 0 & 0 & 0 & J_{\beta_4} \end{bmatrix} \tag{3.11}$$

Referring to Eq. (3.2), one has

$$J_{\beta_j} = \frac{\partial \theta_i}{\partial \beta_j} = \delta_j \tag{3.12}$$

with $i = 1$ if $j = 0$ and $i = j$ otherwise. It may be observed from Eq. (3.11) that $\dot{\theta}_1$ is calculated as an average of the contributions of $\dot{\rho}_0$ and $\dot{\rho}_1$ due to the swing joint's kinematic redundancy.[1] Finally,

[1] Note that $\dot{\theta}_1$ could just as well be computed using only one of either $\dot{\rho}_0$ or $\dot{\rho}_1$.

the Cartesian velocity of the breaker is obtained as $\dot{\mathbf{x}} = \mathbf{J}_\theta \dot{\boldsymbol{\theta}}$ with

$$\mathbf{J}_\theta = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\theta}} = \begin{bmatrix} -\mathrm{s}_1(a_1 + a_2\mathrm{c}_2 + a_3\mathrm{c}_{23} + a_4\mathrm{c}_{234}) & -\mathrm{c}_1(a_2\mathrm{s}_2 + a_3\mathrm{s}_{23} + a_4\mathrm{s}_{234}) & -\mathrm{c}_1(a_3\mathrm{s}_{23} + a_4\mathrm{s}_{234}) & -\mathrm{c}_1(a_4\mathrm{s}_{234}) \\ \mathrm{c}_1(a_1 + a_2\mathrm{c}_2 + a_3\mathrm{c}_{23} + a_4\mathrm{c}_{234}) & -\mathrm{s}_1(a_2\mathrm{s}_2 + a_3\mathrm{s}_{23} + a_4\mathrm{s}_{234}) & -\mathrm{s}_1(a_3\mathrm{s}_{23} + a_4\mathrm{s}_{234}) & -\mathrm{s}_1(a_4\mathrm{s}_{234}) \\ 0 & a_2\mathrm{c}_2 + a_3\mathrm{c}_{23} + a_4\mathrm{c}_{234} & a_3\mathrm{c}_{23} + a_4\mathrm{c}_{234} & a_4\mathrm{c}_{234} \\ 0 & 1 & 1 & 1 \end{bmatrix} \tag{3.13}$$

which may be obtained from Eqs. (3.6) and (3.7).

In order to convert $\dot{\mathbf{x}}$ into $\dot{\boldsymbol{\rho}}$, one may first observe that $\dot{\boldsymbol{\theta}} = \mathbf{J}_\theta^{-1}\dot{\mathbf{x}}$ where the inverse of $\mathbf{J}_\theta$ exists so long as the rockbreaker is operating within its workspace. Afterwards, one has $\dot{\boldsymbol{\beta}} = \mathbf{J}_\beta^* \dot{\boldsymbol{\theta}}$ with

$$\mathbf{J}_\beta^* = \begin{bmatrix} 1/J_{\beta_0} & 0 & 0 & 0 \\ 1/J_{\beta_1} & 0 & 0 & 0 \\ 0 & 1/J_{\beta_2} & 0 & 0 \\ 0 & 0 & 1/J_{\beta_3} & 0 \\ 0 & 0 & 0 & 1/J_{\beta_4} \end{bmatrix} \tag{3.14}$$

where it may be observed that $\mathbf{J}_\beta^* \neq \mathbf{J}_\beta^{-1}$. Finally, the actuator velocities may be obtained as $\dot{\boldsymbol{\rho}} = \mathbf{J}_\rho^{-1}\dot{\boldsymbol{\beta}}$. Overall, one obtains

$$\dot{\boldsymbol{\rho}} = \mathbf{J}\dot{\mathbf{x}} \tag{3.15}$$

where $\mathbf{J} = \mathbf{J}_\rho^{-1}\mathbf{J}_\beta^*\mathbf{J}_\theta^{-1}$.

3.5 Workspace analysis

Although the feasibility of rockbreaker trajectories may be verified from the solution to the IKP, the computation of its workspace provides for a useful visualization tool in the context of trajectory planning. Given the nature of the rockbreaker's architecture, the planar workspace for the case where $\theta_1 = 0$ is first determined as shown in Figure 3.3. Potential workspace boundaries are obtained by sequentially setting the angle of two joints among those located at points R_2, R_3 and R_4 to either of their operating range limits (*i.e.* $\theta_{i_{\min}} \leq \theta_i \leq \theta_{i_{\max}}$) and then sweeping the remaining joint through

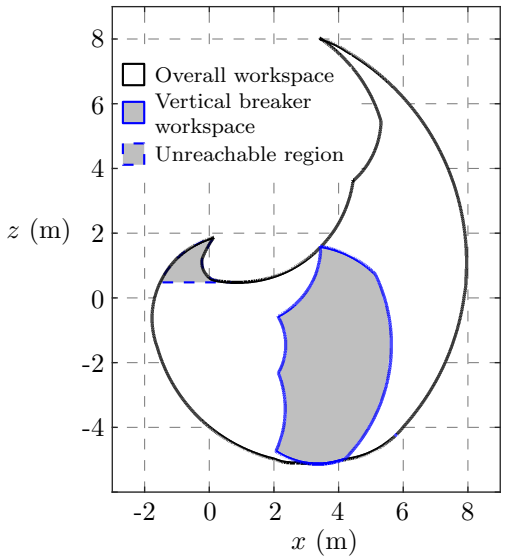


Figure 3.3: Rockbreaker workspace for the case where $\theta_1 = 0$.

its own operating range. The corresponding curve representing the potential workspace boundary in Cartesian space is then obtained through the solution of the DKP. A set of potential workspace boundaries is obtained by repeating this procedure for all possible pairs of revolute joints among those located at points R_2, R_3 and R_4. Simple geometrical tests may then be used to identify which of the curves represent actual workspace boundaries. In order to visualize the rockbreaker's three-dimensional workspace, one may imagine the planar workspace shown in Figure 3.3 being swept through the range of operation of the first revolute joint (*i.e.* R_1). Also shown in Figure 3.3 is the region of the workspace that is attainable by the rockbreaker with a vertical breaker orientation (*i.e.* where $\phi = 270$ degrees). This particular area of the workspace is identified due to the fact that rock breaking is typically accomplished more easily with the system in such an orientation (although this depends on the orientation of the surface of the rock needing to be fragmented).

3.6 Flow rate analysis

As described in Section 3.1, the rockbreaker's motion is constrained by maximum admissible flow rates $Q_{i_{\max}}$ through each of its proportional valves as well as the maximum flow rate $Q_{p_{\max}}$ of its hydraulic pump. Conversion between the rockbreaker's hydraulic actuator velocities and corresponding valve flow rates is required to verify the satisfaction of these flow rate limit constraints. Given $\dot{\boldsymbol{\rho}}$,

the flow rates in the proportional valves may be obtained as $\mathbf{Q}_v = \mathbf{J}_v|\dot{\boldsymbol{\rho}}|$ with

$$\mathbf{J}_v = \begin{bmatrix} A_0 & A_1 & 0 & 0 & 0 \\ 0 & 0 & A_2 & 0 & 0 \\ 0 & 0 & 0 & A_3 & 0 \\ 0 & 0 & 0 & 0 & A_4 \end{bmatrix} \tag{3.16}$$

and where $\mathbf{Q}_v = [Q_1, Q_2, Q_3, Q_4]^T$ and $|\dot{\boldsymbol{\rho}}| = [|\dot{\rho}_0|, |\dot{\rho}_1|, \ldots, |\dot{\rho}_4|]^T$. Moreover, A_j is the area of the j^{th} actuator on which the hydraulic fluid is acting where $A_j = A_{\text{blind}_j}$ when $\dot{\rho}_j > 0$ and $A_j = A_{\text{rod}_j}$ when $\dot{\rho}_j < 0$. A_{rod_j} and A_{blind_j}, for their part, are the rod side and blind side surface areas of the j^{th} actuator's piston, respectively. As may be observed from the preceding developments, flow rates are always considered to be positive here as they must collectively be supplied by the hydraulic pump regardless of their direction of flow through an actuator. In fact, the required pump flow rate is simply obtained as

$$Q_p = \sum_{i=1}^{4} Q_i \tag{3.17}$$

such that an overall flow rate vector $\mathbf{Q} = [Q_1, Q_2, Q_3, Q_4, Q_p]^T$ may be obtained as $\mathbf{Q} = \mathbf{J}_p\mathbf{Q}_v$ with

$$\mathbf{J}_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.18}$$

Alternatively, one may also write

$$\mathbf{Q} = \mathbf{J}_q|\dot{\boldsymbol{\rho}}| \tag{3.19}$$

with $\mathbf{J}_q = \mathbf{J}_p\mathbf{J}_v$.

Chapter 4

Dynamic modeling

While previous works have consistently used simplified dynamic models of large hydraulic machinery, to the author's knowledge none have validated such simplifications through dynamic simulations. The research presented in this chapter seeks to mitigate this shortcoming by developing a detailed dynamic model of a 4-DoF rockbreaker. The scope of the work presented within this chapter includes the dynamic modeling of the rockbreaker mechanism (*i.e.* links, joints and actuators) but excludes the hydraulic system used for the actuation. Furthermore, it is assumed that the rockbreaker is anchored to a stationary pedestal while interactions with its environment (*e.g.* rock breaking) are not considered. In order to reduce the complexity of the dynamic model and ease the implementation of an adaptive control algorithm, the neglect of some terms within the dynamic model will be explored and the corresponding effects will be quantified. The possibility of neglecting terms within the dynamic model is cited by [13], who states that the effects of the Coriolis and centrifugal terms of the dynamic model of excavators are negligible due to the system's relatively low velocities. The goal is to identify a simplified dynamic model of a rockbreaker that represents the actual system with sufficient accuracy while allowing for an easier parameterization to be used in an adaptive controller, similar to the one presented in [34]. It is expected that this would lead to corresponding reductions of the time required within a real-time control algorithm to compute the dynamic model although this is not validated in this work.

The remainder of this chapter is organized as follows. A more in depth kinematic analysis of the system is provided in Section 4.1, with regards to the computation of the positions of centres of

masses for the rigid bodies within the rockbreaker system. The dynamic modeling of the rockbreaker is detailed in Section 4.2. An analysis of the relative contributions of various terms within the dynamic model on the rockbreaker's joint torques for sample trajectories are discussed in Section 4.3 while some conclusions are presented in Section 4.4.

4.1 Kinematic analysis

Although a general kinematic analysis of the rockbreaker was provided in Chapter 3, positions of the centres of mass of all rigid bodies along with the latter's angular velocities are required for the development of a dynamic model. Referring to Figure 4.1, the centres of mass of the links as well as those of each hydraulic actuator barrel and piston assembly are labeled as C_{l_i}, C_{b_j} and C_{p_j}, respectively. It should be noted that, although they may appear in-line with the prismatic actuators' axes (defined as the lines joining nodes A_j and B_j) in Figure 4.1, the centres of mass of the actuator barrels and piston assemblies are generally offset from these axes, which adds some complexity to the dynamic model.

The position of the centre of mass of the i^{th} link with respect to the $(i+1)^{\text{th}}$ reference frame (which is attached to this link) is defined by the vector $\mathbf{s}_{l_i}$ (see Figure 4.1(d)). The position of the centre of mass C_{l_i} of the i^{th} link ($i = 1, 2, 3, 4$) with respect to the origin of the $X_1Y_1Z_1$ reference frame may thus be described as:

$$[\bar{\mathbf{c}}_{l_i}]_1 = \sum_{n=1}^{i} \mathbf{R}_n^1 [\mathbf{a}_n]_n + [\mathbf{s}_{l_i}]_1 = [\mathbf{a}_1]_1 + \mathbf{R}_2^1 [\mathbf{a}_2]_2 + \cdots + \mathbf{R}_i^1 [\mathbf{a}_i]_i + \mathbf{R}_{i+1}^1 [\mathbf{s}_{l_i}]_{i+1} \tag{4.1}$$

As for the absolute angular velocity of the i^{th} link, it may be obtained as:

$$[\boldsymbol{\omega}_{l_i}]_1 = \sum_{n=1}^{i} \mathbf{R}_n^1 \left[0, 0, \dot{\theta}_n\right]^T \tag{4.2}$$

where $\mathbf{R}_1^1 = \mathbf{1}_{3\times 3}$ is the identity matrix and where $\mathbf{R}_i^1 = \prod_{n=2}^{i} \mathbf{R}_n^{n-1}$. A similar approach is used to define the positions of the centres of mass and the angular velocities of the actuator barrels and piston assemblies. The position of the centre of mass of the j^{th} actuator barrel measured from node A_j is defined as $\mathbf{s}_{b_j}$. Likewise, the position of the centre of mass of the j^{th} actuator piston assembly, measured from node B_j, is defined as $\mathbf{s}_{p_j}$. Both of these vectors are constant in a frame $X_j'Y_j'Z_j'$

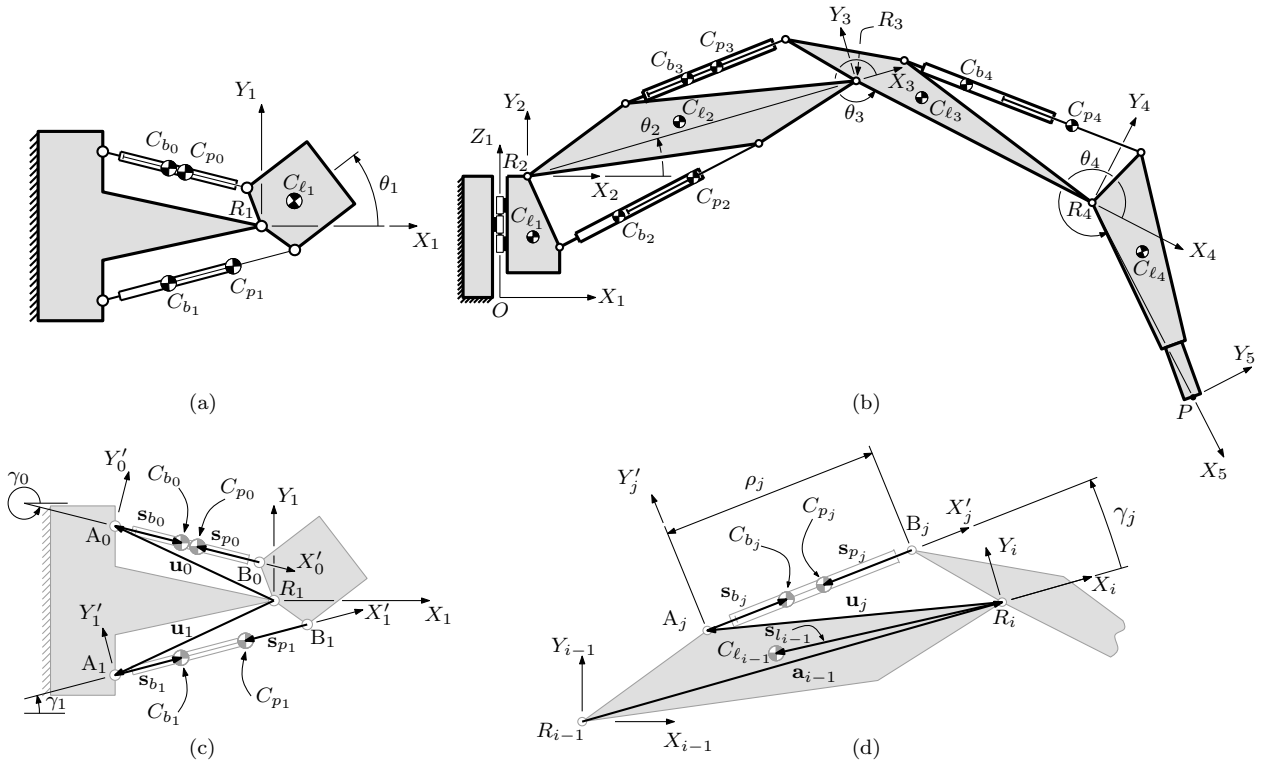


Figure 4.1: Schematic representation of the rockbreaker with parameter definitions: (a) and (c) top view of base and swing post, (b) side view of complete rockbreaker and (d) partial side view of rockbreaker.

which is defined as having its X'_j axis directed from A_j to B_j while its Z'_j axis is parallel to the axis of the passive revolute joint at A_j as shown in Figures 4.1(c) and (d). The rotation matrix bringing the $X_iY_iZ_i$ frame parallel to the $X'_jY'_jZ'_j$ is:

$$\mathbf{R}^i_{j'} = \begin{bmatrix} \cos\gamma_j & -\sin\gamma_j & 0 \\ \sin\gamma_j & \cos\gamma_j & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

where γ_j is measured from the X_i axis to the X'_j axis. Finally, vector $\mathbf{u}_j$ is defined as being directed from R_i to A_j where $i = j$ when $j = 1, 2, 3, 4$ and $i = j + 1$ when $j = 0$. It is noted that this vector is constant in the $X_iY_iZ_i$ frame. With these definitions, the positions of the centres of mass of the j^{th} actuator barrel and piston assembly, for the case where $j = 0, 1$, are obtained as:

$$[\bar{\mathbf{c}}_{b_j}]_1 = [\mathbf{u}_j]_1 + \mathbf{R}^1_{j'}[\mathbf{s}_{b_j}]_{j'} \tag{4.4}$$

$$[\bar{\mathbf{c}}_{p_j}]_1 = [\mathbf{u}_j]_1 + \mathbf{R}^1_{j'}\big(\rho_j[\mathbf{i}_{j'}] + [\mathbf{s}_{p_j}]_{j'}\big) \tag{4.5}$$

where $[\cdot]_{j'}$ is a vector expressed in the $X'_jY'_jZ'_j$ reference frame, $\mathbf{i}_{j'}$ is a unit vector directed along the X'_j axis and where $\mathbf{R}^1_{j'} = \left(\prod_{n=2}^{j} \mathbf{R}^{n-1}_n\right)\mathbf{R}^j_{j'}$. Similarly, the positions of the centres of mass of the j^{th} actuator barrel and piston assembly, for cases where $j = 2, 3, 4$, may be found as:

$$\begin{aligned}[\bar{\mathbf{c}}_{b_j}]_1 &= \sum_{n=1}^{j-1} \mathbf{R}^1_n[\mathbf{a}_n]_1 + [\mathbf{u}_j]_1 + [\mathbf{s}_{b_j}]_1 \\ &= [\mathbf{a}_1]_1 + \mathbf{R}^1_2[\mathbf{a}_2]_2 + \cdots + \mathbf{R}^1_{j-1}[\mathbf{a}_{j-1}]_{j-1} + \mathbf{R}^1_j[\mathbf{u}_j]_j + \mathbf{R}^1_{j'}[\mathbf{s}_{b_j}]_{j'}\end{aligned} \tag{4.6}$$

$$\begin{aligned}[\bar{\mathbf{c}}_{p_j}]_1 &= \sum_{n=1}^{j-1} \mathbf{R}^1_n[\mathbf{a}_n]_n + [\mathbf{u}_j]_1 + \rho_j[\mathbf{i}_{j'}]_1 + [\mathbf{s}_{p_j}]_1 \\ &= [\mathbf{a}_1]_1 + \mathbf{R}^1_2[\mathbf{a}_2]_2 + \cdots + \mathbf{R}^1_{j-1}[\mathbf{a}_{j-1}]_{j-1} + \mathbf{R}^1_j[\mathbf{u}_j]_j + \mathbf{R}^1_{j'}\left(\rho_j[\mathbf{i}_{j'}]_{j'} + [\mathbf{s}_{p_j}]_{j'}\right)\end{aligned} \tag{4.7}$$

Finally, the angular velocities of the j^{th} actuator's barrel and piston assembly, measured with respect to ground and expressed in frame $X_1Y_1Z_1$, can be computed as:

$$[\boldsymbol{\omega}_{b_j}]_1 = [\boldsymbol{\omega}_{p_j}]_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\gamma}_j \end{bmatrix} \quad (j = 0, 1) \tag{4.8}$$

$$[\boldsymbol{\omega}_{b_j}]_1 = [\boldsymbol{\omega}_{p_j}]_1 = [\boldsymbol{\omega}_{l_{j-1}}]_1 + \prod_{n=2}^{j} \mathbf{R}^{n-1}_n \begin{bmatrix} 0 \\ 0 \\ \dot{\gamma}_j \end{bmatrix} \quad (j = 2, 3, 4) \tag{4.9}$$

4.2 Development of dynamic models

To develop the dynamic model for the rockbreaker, the Lagrange-Euler method was used with the joint angles θ_i as generalized coordinates, *i.e.*

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{\theta}_i} - \frac{\partial T}{\partial \theta_i} + \frac{\partial U}{\partial \theta_i} = \tau_i, \quad i = 1, 2, 3, 4 \tag{4.10}$$

where T and U represent the rockbreaker's total kinetic and potential energies, respectively. The kinetic and potential energies can be decomposed into contributions from the links (l), actuator barrels (b) and actuator pistons (p) such that $T = T_l + T_b + T_p$ and $U = U_l + U_b + U_p$ with:

$$T_l = \sum_{i=1}^{4} \left[\frac{1}{2} m_{l_i} \dot{\bar{\mathbf{c}}}^T_{l_i} \dot{\bar{\mathbf{c}}}_{l_i} + \frac{1}{2} \boldsymbol{\omega}^T_{l_i} \mathbf{R}^1_{i+1} \bar{\mathbf{I}}_{l_i} \left(\mathbf{R}^1_{i+1}\right)^T \boldsymbol{\omega}_{l_i}\right], \quad U_l = -g\mathbf{e}^T_g \left(\sum_{i=1}^{4} m_{l_i} \bar{\mathbf{c}}_{l_i}\right) \tag{4.11}$$

$$T_b = \sum_{j=0}^{4} \left[\frac{1}{2} m_{b_j} \dot{\bar{\mathbf{c}}}_{b_j}^T \dot{\bar{\mathbf{c}}}_{b_j} + \frac{1}{2} \boldsymbol{\omega}_{b_j}^T \mathbf{R}_{j'}^1 \bar{\mathbf{I}}_{b_j} \left(\mathbf{R}_{j'}^1\right)^T \boldsymbol{\omega}_{b_j} \right], \quad U_b = -g \mathbf{e}_g^T \left(\sum_{j=0}^{4} m_{b_j} \bar{\mathbf{c}}_{b_j} \right) \tag{4.12}$$

$$T_p = \sum_{j=0}^{4} \left[\frac{1}{2} m_{p_j} \dot{\bar{\mathbf{c}}}_{p_j}^T \dot{\bar{\mathbf{c}}}_{p_j} + \frac{1}{2} \boldsymbol{\omega}_{p_j}^T \mathbf{R}_{j'}^1 \bar{\mathbf{I}}_{p_j} \left(\mathbf{R}_{j'}^1\right)^T \boldsymbol{\omega}_{p_j} \right], \quad U_p = -g \mathbf{e}_g^T \left(\sum_{j=0}^{4} m_{p_j} \bar{\mathbf{c}}_{p_j} \right) \tag{4.13}$$

where $i = 1, 2, 3, 4$ and $j = 0, 1, \ldots, 4$. In these equations, m_{l_i}, m_{b_j} and m_{p_j} represent the masses of the i^{th} link, j^{th} actuator barrel and j^{th} actuator piston assembly, respectively. Meanwhile, $\dot{\bar{\mathbf{c}}}_{l_i}$, $\dot{\bar{\mathbf{c}}}_{b_j}$ and $\dot{\bar{\mathbf{c}}}_{p_j}$ represent the absolute velocities of each rigid body's centre of mass, which are obtained through the time derivative of Eqs. (4.1), (4.4), (4.5), (4.6) and (4.7), *i.e.* the position-level equations of the terms. The inertia matrix of the i^{th} link measured in a frame parallel to the $X_{i+1}Y_{i+1}Z_{i+1}$ frame with origin at C_{l_i} is represented as $\bar{\mathbf{I}}_{l_i}$. Similarly, the inertia matrices of the j^{th} actuator barrel and piston assembly measured in reference frames parallel to the $Y_j'Y_j'Z_j'$ frame with origins at C_{b_j} and C_{p_j}, respectively, are represented as $\bar{\mathbf{I}}_{b_j}$ and $\bar{\mathbf{I}}_{p_j}$. Finally, $g = 9.80655$ m/s^2 is the gravitational acceleration and $\mathbf{e}_g$ is a unit vector parallel to the gravitational field (*i.e.* $[\mathbf{e}_g]_1 = [0, 0, -1]^T$). Using the Lagrange-Euler method, the complete dynamic model can then be obtained as:

$$\boldsymbol{\tau} = \mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{v}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{g}(\boldsymbol{\theta}) \tag{4.14}$$

where the generalized inertia matrix $\mathbf{M}(\boldsymbol{\theta})$, the vector of the Coriolis and centrifugal effects $\mathbf{v}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ and the vector of gravitational effects $\mathbf{g}(\boldsymbol{\theta})$ can be divided into two parts: one due to contributions from the main links of the rockbreaker (labeled *link*) and the other due to contributions from the hydraulic actuators (both barrels and piston assemblies, labeled *act*) such that:

$$\mathbf{M}(\boldsymbol{\theta}) = \mathbf{M}_{link}(\boldsymbol{\theta}) + \mathbf{M}_{act}(\boldsymbol{\theta}) \tag{4.15}$$

$$\mathbf{v}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \mathbf{v}_{link}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{v}_{act}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \tag{4.16}$$

$$\mathbf{g}(\boldsymbol{\theta}) = \mathbf{g}_{link}(\boldsymbol{\theta}) + \mathbf{g}_{act}(\boldsymbol{\theta}) \tag{4.17}$$

Moreover, $\boldsymbol{\tau} = [\tau_1, \tau_2, \tau_3, \tau_4]^T$ is the vector of joint torques where τ_i is the torque that is applied indirectly to revolute joint R_i by the hydraulic actuators.

The dynamic model presented in Eq. (4.14) was developed using symbolic math software though detailed expressions for the different terms within the model are not presented here due to their complex and extensive nature. Validation of the analytical model was achieved by comparing the

results with a model developed using the Simscape Multibody toolbox in MATLAB/SIMULINK®. Computed joint torques from both models were found to match to a precision of 1×10^{-6} N·m, which corresponds to the precision of the model parameters used in the simulations. In order to improve the computational efficiency and reduce model complexity, simplified versions of the dynamic model are introduced as follows:

$$\boldsymbol{\tau}_I = \mathbf{M}_{link}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{v}_{link}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{g}(\boldsymbol{\theta}) \tag{4.18}$$

$$\boldsymbol{\tau}_{II} = \mathbf{M}_{link}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{v}_{link}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{g}_{link}(\boldsymbol{\theta}) \tag{4.19}$$

$$\boldsymbol{\tau}_{III} = \mathbf{M}_{link}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{g}(\boldsymbol{\theta}) \tag{4.20}$$

In all cases, the contributions of the hydraulic actuators to $\mathbf{M}(\boldsymbol{\theta})$ and $\mathbf{v}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ have been neglected although their contribution to $\mathbf{g}(\boldsymbol{\theta})$ has been maintained in the models described by Eqs. (4.18) and (4.20). These simplified models will be compared to the full dynamic model in terms of joint torques corresponding to prescribed rockbreaker trajectories. The objective is to identify a dynamic model of reduced complexity that generates joint torques sufficiently close to those obtained from the full model. The incorporation of a model-based component with a high accuracy into the control system will reduce the feedback control gains and will contribute to maintain a reasonable stability region around the equilibrium point [66]. It is estimated that joint torque errors less than approximately ±10% between a simplified dynamic model and the full model in Eq. (4.14) would contribute to meeting this objective. However, this would need to be verified by implementing a model-based controller on the actual rockbreaker.

4.3 Dynamic simulation results

The use of the dynamic model in an adaptive control scheme requires it to be expressed in parameterized form, which would be a complex task if the complete dynamic model were used. For this reason, simulations are performed using the simplified dynamic models that were introduced in Eqs. (4.18) – (4.20) and the resulting joint torques are compared to those obtained with the full dynamic model. The goal is to identify a simplified dynamic model whose deviation from the full model is small enough to allow a suitable semi-automated system performance through the use of a nonlinear dynamic model-based controller.

The simplified dynamic models are compared to the full dynamic model through the resulting joint torques (*i.e.* the torques applied indirectly to joints R_i by the hydraulic actuators). On one hand, the absolute joint torque error $\Delta\tau_{k,i}$ at the i^{th} joint is computed as:

$$\Delta\tau_{k,i} = |\tau_{k,i} - \tau_i| \tag{4.21}$$

where τ_i is the torque obtained from the full dynamic model (from Eq. (4.14)) and $\tau_{k,i}$ ($k = I, II, III$) is the one obtained from the k^{th} simplified dynamic model (Eqs. (4.18) – (4.20)). On the other hand, the relative joint torque error, defined as:

$$\varepsilon_{k,i} = (\Delta\tau_{k,i}/\tau_i) \cdot 100 \quad (\%) \tag{4.22}$$

will also be computed and analyzed over the simulated trajectories. The relative joint torque error is useful since it provides some perspective of the scale of the discrepancies between the different dynamic models. However, in some instances, the relative error becomes very large due to the fact that the actual joint torques (*i.e.* τ_i) are very small or equal to zero. In these cases, the absolute joint torque error, compared to the magnitude of the corresponding joint torques obtained from the full dynamic model, gives a better indication of a simplified dynamic model's adequacy.

Simulations are based on predefined trajectories obtained with a trajectory planning algorithm. The algorithm used to generate these trajectories utilizes a decoupled trajectory planning approach (independent swing and planar arm motion) as described in Chapter 5 with the flow optimization presented in Chapter 6. This leads to trajectories meeting the following criteria:

- The flow rates to the hydraulic actuators do not exceed the allowable flow rates through the valves.
- The overall sum of flow rates to the hydraulic actuators does not exceed the hydraulic pump's maximum flow rate.
- Trajectories ensure smooth ramp-up and ramp-down of joint velocities.
- Trajectories ensure that at least one hydraulic actuator is operating at its maximum available flow rate at all times (to allow for the fastest motion possible).
- Trajectories maintain the breaker at an orientation as close as possible to vertical at all times (*i.e.* $\phi = 270$ degrees). Note that this is an operational requirement for the targeted application.

- Trajectories respect the mechanical limits of all joints.

The parameter values that were used for all simulations were taken from an actual rockbreaker (obtained through CAD models and hydraulic actuator data sheets) and are provided in Appendix A. It should be noted that, in practice, the actual system may not exhibit identical physical properties (due to tolerances in the manufacturing process, variations in physical properties and unmodeled effects) as the ones provided here[1]. Two trajectories, referred to as trajectory 1 and trajectory 2 and illustrated in Figure 4.2, were used to compare the simplified dynamic models to the full model. These trajectories were selected as they provided a good representation of typical rockbreaker motions while providing a broad range of motion needed for a thorough analysis of the system dynamics. In what follows, $\mathbf{p}_{I_i}$ and ϕ_{I_i} are the initial breaker position and orientation for the i^{th} trajectory, respectively, with $\mathbf{p}_{F_i}$ and ϕ_{F_i} being their equivalents for the final breaker position and orientation.

- **Trajectory 1:** The tip of the breaker (*i.e.* P) is displaced from $\mathbf{p}_{I_1} = [6.50,\ -2.50,\ -0.50]^T$ m to $\mathbf{p}_{F_1} = [2.55,\ 2.50,\ -0.50]^T$ m while its orientation changes from $\phi_{I_1} = 288$ degrees to $\phi_{F_1} = 270$ degrees. This is a motion where the breaker moves between diagonally opposed corner openings of the grizzly within the work plane located above the latter (which can be visualized using Figure 3.1).

- **Trajectory 2:** The tip of the breaker (*i.e.* P) is displaced from $\mathbf{p}_{I_2} = [4.50,\ 0.00,\ -0.50]^T$ m to $\mathbf{p}_{F_2} = [0.65,\ 4.95,\ 0.00]^T$ m while its breaker is maintained in a vertical orientation such that $\phi_{I_2} = \phi_{F_2} = 270$ degrees. This is a motion where the breaker moves from a central location above the grizzly to an arbitrary home position for the rockbreaker where it is parked when not in use.

The simulation results for trajectory 1 in terms of the full dynamic model's computed joint torque τ_i as well as the absolute and relative joint torque errors can be seen in Figures. 4.3 to 4.6. It can be observed that the best approximation of the full dynamic model is represented by $\boldsymbol{\tau}_I$. Its proximity to the full dynamic model suggests that the $\mathbf{M}_{act}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}}$ and $\mathbf{v}_{act}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ terms have minimal impact

[1] The resulting effects of poor parameter values could lead to poor performance of a model-based control algorithm, however this may be mitigated when using an adaptive controller (ensuring parameter estimates converge towards their optimal values), as previously mentioned.

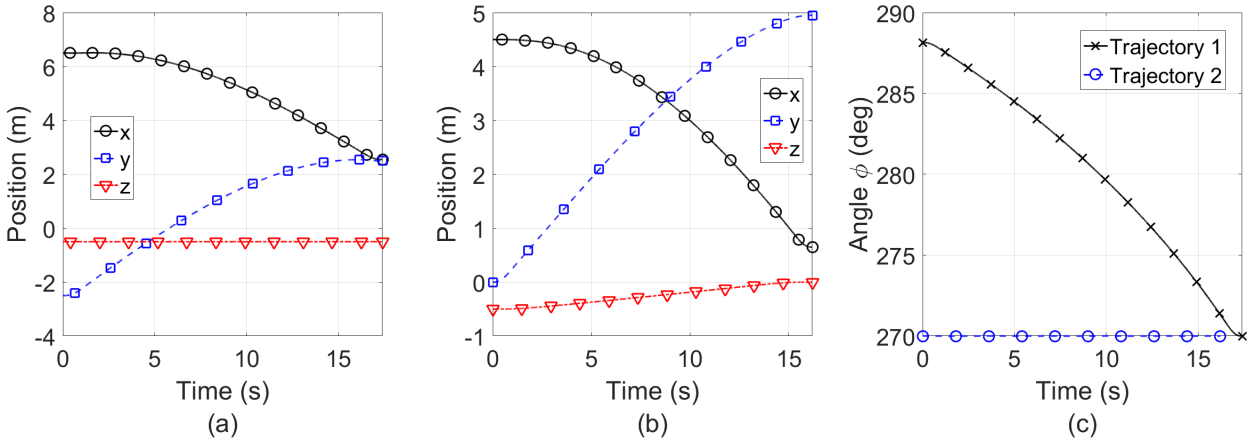


Figure 4.2: Cartesian trajectories of rockbreaker used for dynamic simulations: (a) and (b) Cartesian position coordinates for trajectories 1 and 2, respectively, and (c) orientation ϕ for trajectories 1 and 2.

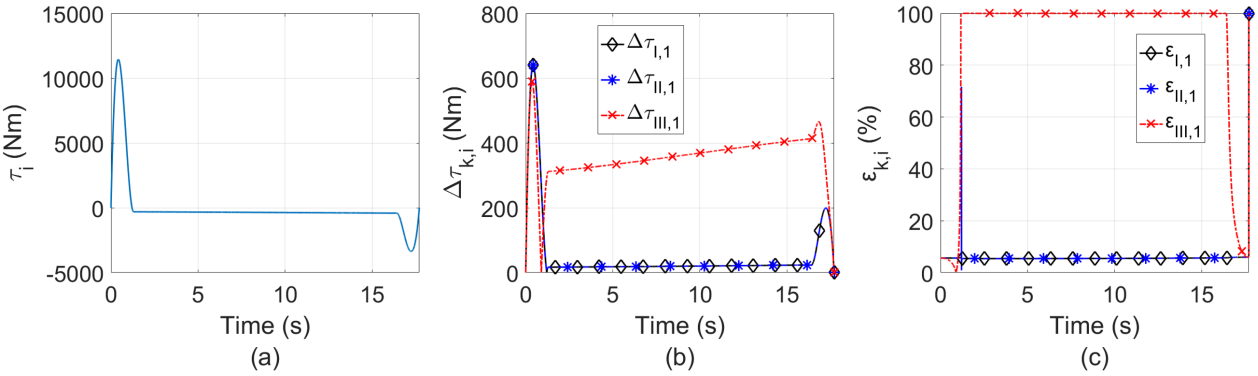


Figure 4.3: Simulation results of torque at R_1 for trajectory 1: (a) computed joint torque using the full dynamic model, (b,c) absolute and relative joint torque errors, respectively, based on simplified dynamic models.

on the system's dynamics. However, this model remains highly complex due to the $\mathbf{v}_{link}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ term and is therefore less than desirable for its implementation in an adaptive control scheme. Furthermore, it may be observed in Figures 4.3 (b) and (c) that $\Delta\tau_{I,1} = \Delta\tau_{II,1}$ and $\varepsilon_{I,1} = \varepsilon_{II,1}$. This may be explained by the fact that the models differ only with regards to gravitational effects which have no effect on the R_1 joint torques. Looking at the plots of $\varepsilon_{I,i}$ and $\varepsilon_{II,i}$ in Figures. 4.4 to 4.6, the considerable influence of the gravitational effect of the actuators $\mathbf{g}_{act}(\boldsymbol{\theta})$ on the joint torques, accounting for as much as 8% of the joint torques computed from the full dynamic model, suggests that Eq. (4.19) is not a suitable simplified dynamic model. Lastly, as can be observed in

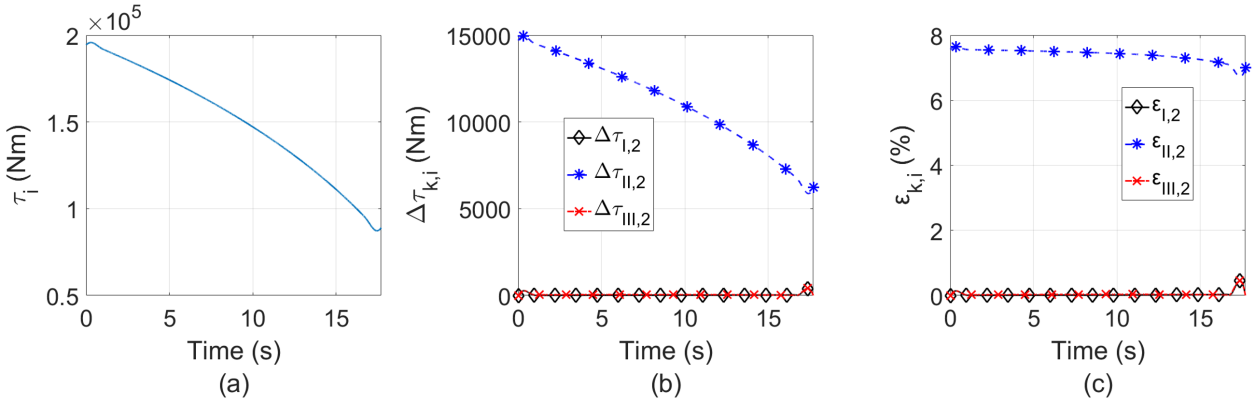


Figure 4.4: Simulation results of torque at R_2 for trajectory 1: (a) computed joint torque using the full dynamic model, (b,c) absolute and relative joint torque errors, respectively, based on simplified dynamic models.

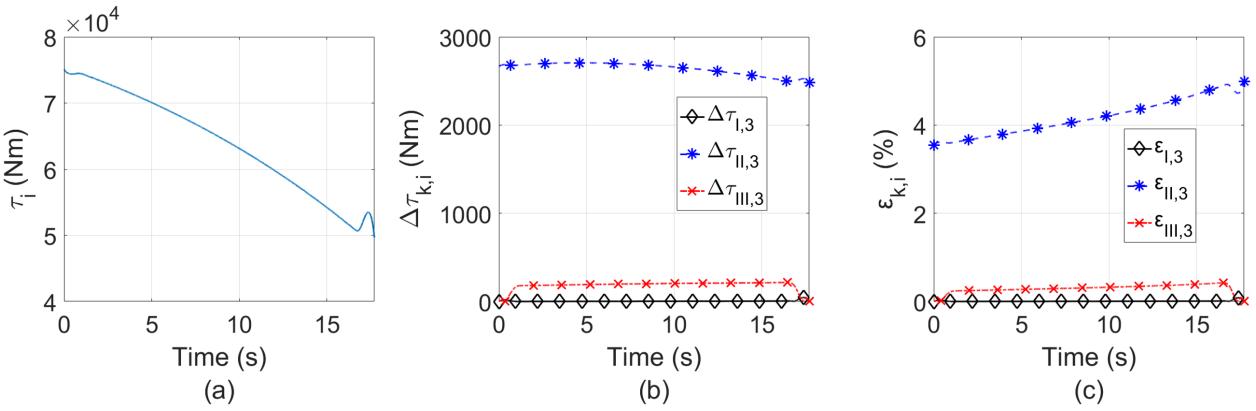


Figure 4.5: Simulation results of torque at R_3 for trajectory 1: (a) computed joint torque using the full dynamic model, (b,c) absolute and relative joint torque errors, respectively, based on simplified dynamic models.

Figure 4.3(c), $\varepsilon_{III,1}$ reaches 100% in the central portion of the simulated trajectory (where $\ddot{\theta}_i = 0$) due to the complete omission of the $\mathbf{v}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ term in $\boldsymbol{\tau}_{III}$ and the fact that $\mathbf{g}(\boldsymbol{\theta})$ has no bearing on the swing joint at R_1. However, the relative errors of $\boldsymbol{\tau}_{III}$ for the subsequent joints ($\varepsilon_{III,2}$, $\varepsilon_{III,3}$ and $\varepsilon_{III,4}$ as seen in Figures. 4.4 to 4.6) do not exceed 1%.

The simulation results for trajectory 2 are shown in Figures. 4.7 to 4.10. Many of the observations made for trajectory 1 also apply to the simulation results of trajectory 2. It can be observed from Figure 4.8(c) to Figure 4.10(c) that the relative error $\varepsilon_{II,i}$ is quite considerable and even reaches approximately 10% at times, further validating the claim that the gravitational effects due to the

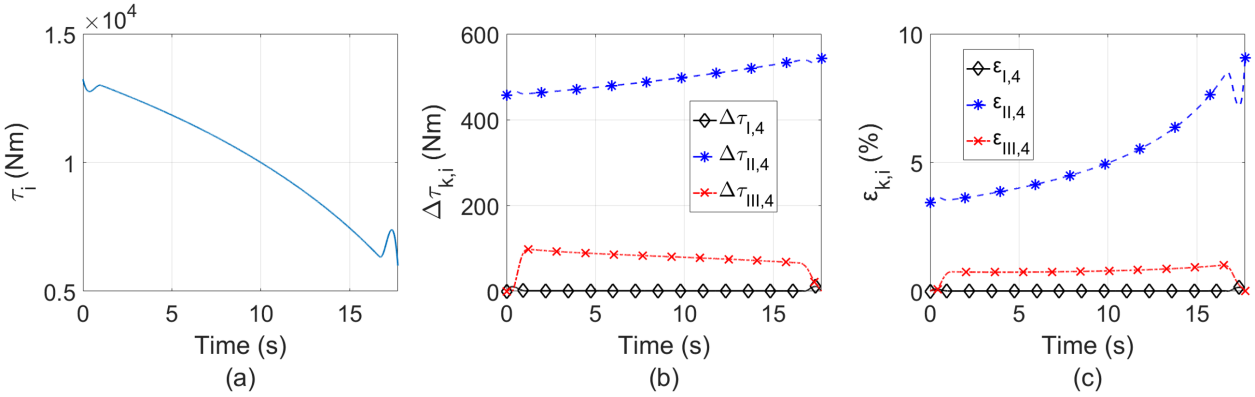


Figure 4.6: Simulation results of torque at R_4 for trajectory 1: (a) computed joint torque using the full dynamic model, (b,c) absolute and relative joint torque errors, respectively, based on simplified dynamic models.

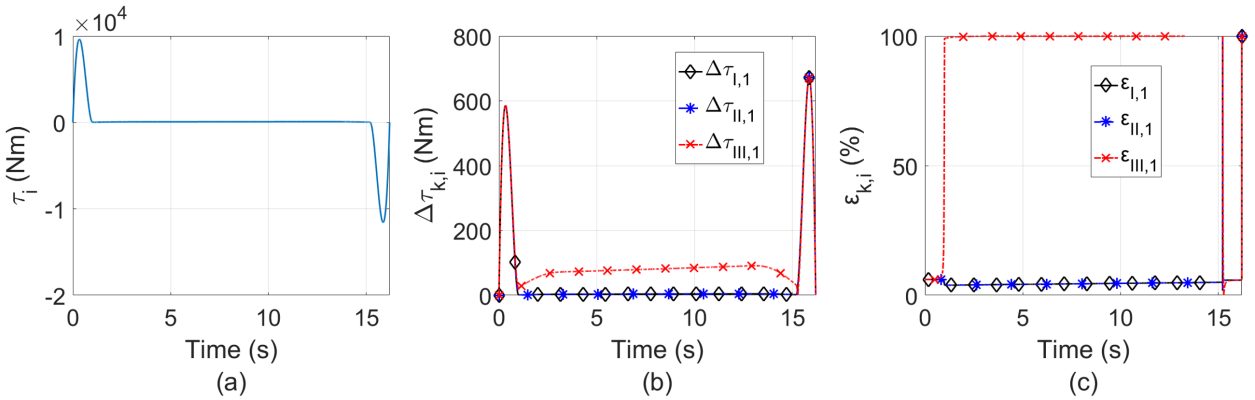


Figure 4.7: Simulation results of torque at R_1 for trajectory 2: (a) computed joint torque using the full dynamic model, (b,c) absolute and relative joint torque errors, respectively, based on simplified dynamic models.

hydraulic actuators are considerable and should not be ignored as was the case in [36–40]. Finally, in the case of trajectory 2, it may be observed that $\varepsilon_{III,2}$, $\varepsilon_{III,3}$ and $\varepsilon_{III,4}$ always remain less than 2%.

From the results obtained from both trajectories 1 and 2, the simplified model given by Eq. (4.20) appears to be an acceptable alternative to the full dynamic model for use in real world applications. Although this model may not seem ideal for the torques at the swing joint (*i.e.* τ_1), it should be noted that the absolute error $\Delta\tau_{III,1}$, which reaches its maximum values during the velocity ramp-up and ramp-down portions of the trajectories (as seen in Figures 4.3(b) and 4.7(b)), remains relatively

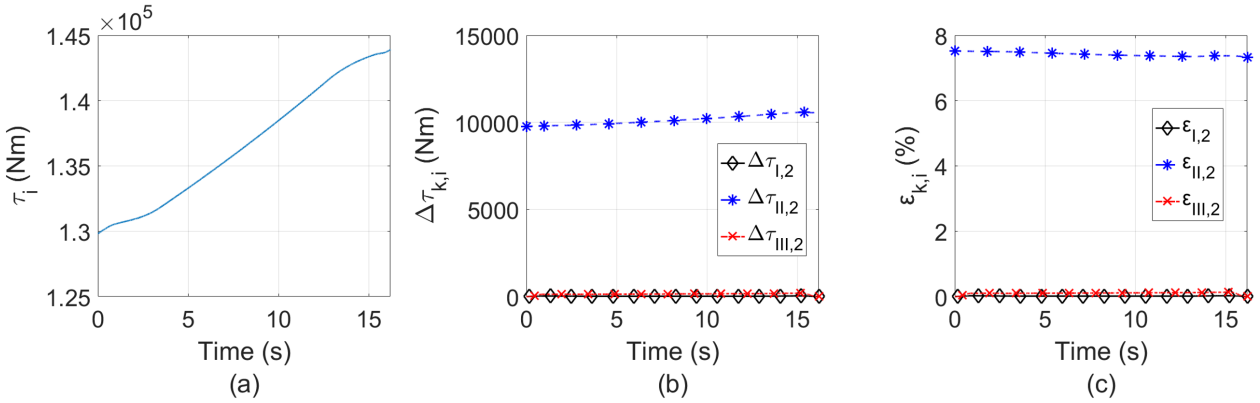


Figure 4.8: Simulation results of torque at R_2 for trajectory 2: (a) computed joint torque using the full dynamic model, (b,c) absolute and relative joint torque errors, respectively, based on simplified dynamic models.

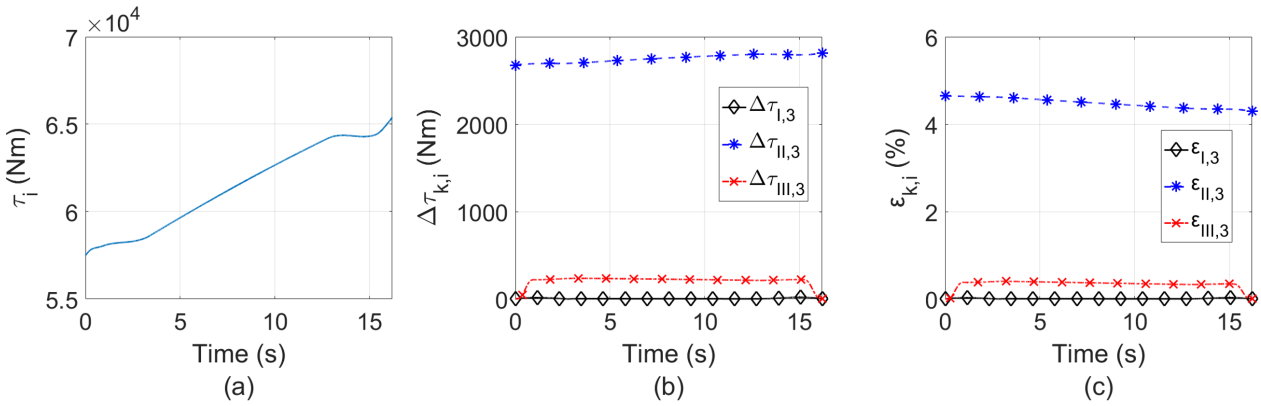


Figure 4.9: Simulation results of torque at R_3 for trajectory 2: (a) computed joint torque using the full dynamic model, (b,c) absolute and relative joint torque errors, respectively, based on simplified dynamic models.

small in magnitude with respect to the actual torques applied to this joint during those motions (as seen in Figures 4.3(a) and 4.7(a)). Furthermore, it is believed that these errors could be mitigated with the addition to Eq. (4.20) of a simplified term replacing the currently omitted $\mathbf{v}_{link}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ term. The parameters within such a simplified term would need to be identified as part of the adaptive control algorithm implementation.

In order to gain a better understanding of the relative importance of the $\mathbf{g}(\boldsymbol{\theta})$ term's contribution to the overall joint torques, the following quantities from Eq. (4.18) are plotted in Figure 4.11 for the case of trajectory 1: $\boldsymbol{\tau}_I^T \mathbf{e}$, $\left[\mathbf{M}_{link}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}}\right]^T \mathbf{e}$, $\left[\mathbf{v}_{link}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\right]^T \mathbf{e}$ and $\left[\mathbf{g}(\boldsymbol{\theta})\right]^T \mathbf{e}$ with $\mathbf{e} = [0, 1, 0, 0]^T$

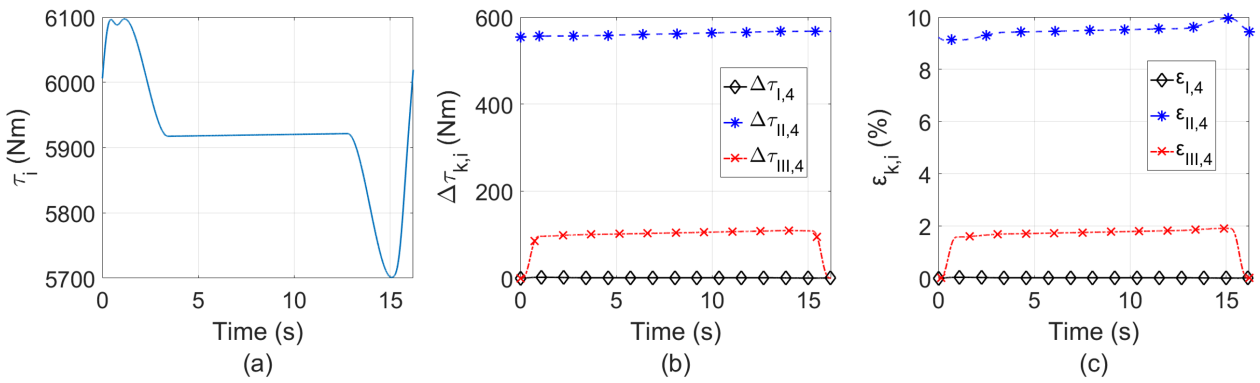


Figure 4.10: Simulation results of torque at R_4 for trajectory 2: (a) computed joint torque using the full dynamic model, (b,c) absolute and relative joint torque errors, respectively, based on simplified dynamic models.

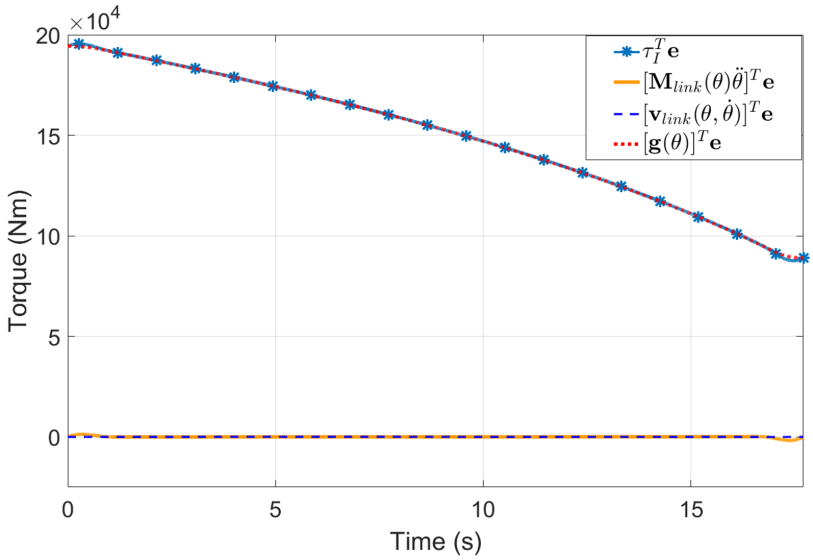


Figure 4.11: Contributions of the individual terms in Eq. (4.18) to the torque at the second joint for trajectory 1.

(these are the individual contributions of the various terms to the torque acting at the rockbreaker's second revolute joint). While these results apply to a single joint in a given trajectory, it is clear that the $\mathbf{g}(\boldsymbol{\theta})$ term represents a very significant portion of the joint torque. As such, the possibility of implementing a further simplified dynamic model whereby only gravitational effects are considered is intriguing. However, one must also keep in mind that gravitational effects are not felt by the rockbreaker's swing joint such that the simplified dynamic model described by Eq. (4.20) remains preferable.

4.4 Conclusions

This chapter provided insight on the dynamic modeling of a hydraulic rockbreaker while investigating potential simplifications to the dynamic model such that it is suitable for use in a model-based controller. It was found that the model proposed in Eq. (4.20), where the vector of Coriolis and centrifugal effects ($\mathbf{v}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$) is entirely neglected and where only the weights of the hydraulic actuators are considered (*i.e.* their contribution to $\mathbf{g}(\boldsymbol{\theta})$) is an attractive alternative to the full dynamic model. It allows for the computation of joint torques τ_2, τ_3 and τ_4 with an error of less than 2% measured relative to the torques obtained from the full dynamic model. The corresponding estimation of τ_1, however, is less accurate owing to the fact that it is not influenced by gravity. The investigation of how this might be addressed through the inclusion of an additional term to compensate for the exclusion of $\mathbf{v}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ without replicating its complexity is left to future work. The dynamic model described in this work did not consider viscous or Coulomb friction at the revolute and prismatic joints. It is believed, however, that friction has a non-negligible effect on the system's dynamics. Also left to future work is the inclusion of friction models with coefficients needing to be identified through an adaptive control scheme.

Chapter 5

Trajectory planning using single point flow rate scaling

As previously mentioned in Chapter 2, although previous research in the development of trajectory planning algorithms for hydraulic rockbreakers and similar machines exists, details regarding the methodology of the algorithms are minimal. Using conventional robotic theory elements regarding trajectory planning algorithms, one may adapt these algorithms to create a trajectory planning algorithm suitable for a semi-autonomous hydraulic rockbreaker. The resulting algorithm should be robust and mitigate issues identified in the literature, ensuring motions are smooth as well as continuous.

The remainder of this chapter is organized as follows. Section 5.1 provides insight on the objectives and constraints for the trajectory planning algorithm while Section 5.2 discusses the computation of the final optimal breaker orientations. A detailed description of the trajectory planning algorithm with single point flow rate scaling is provided in Section 5.3 and sample trajectories computed using this algorithm are presented in Section 5.4. Finally, a discussion of the algorithm's performance and insights on possible improvements are provided in Section 5.5.

5.1 Objectives and constraints

As was stated in Chapter 1, the purpose of developing a semi-autonomous rockbreaker is to free the operator from the continuous rockbreaking operation by automating the motion between rocks that need to be broken. This, in turn, allows the operator to handle multiple rockbreakers simultaneously with the expected benefits of increased productivity and associated cost reductions. Furthermore, the reduction of the operator's involvement reduces the necessary learning curve to operate the rockbreaker effectively. In order to use the system, the operator specifies a rock location on the grizzly through a graphical user interface incorporating a live video feed of the work area. The system then moves the rockbreaker in an automated fashion so that it is located above the rock that needs to be broken. Once the rockbreaker is in position, the user breaks the rock through teleoperation before requesting a movement to a new location. It is assumed that the rockbreaker's automated motion occurs such that the breaker tip remains within a horizontal work plane located above the grizzly, as seen in Figure 3.1, where it will not interact with the surrounding environment or the rock fragments found on the grizzly itself.

A trajectory planning algorithm is required to generate suitable automated motions for the rockbreaker. Inputs to the trajectory planning algorithm consist of the rockbreaker's initial pose $\mathbf{x}_I$ (obtained from joint sensor data through the solution to the rockbreaker's direct kinematic problem) while the final position $\mathbf{p}_F$ is obtained from the operator's input based on an identified rock location combined with knowledge of the work plane elevation. The trajectory planning algorithm outputs the necessary time histories of the actuator displacements, velocities and accelerations to move the rockbreaker between its initial pose to its final pose, which are then sent to a control algorithm for implementation. It should be noted that an auxiliary trajectory planning algorithm[1] is used in a preliminary step to bring the tip of the breaker to the work plane when necessary (*i.e.* if the initial pose of the rockbreaker does not meet this requirement). Furthermore, given a final position $\mathbf{p}_F$, an optimal final breaker orientation (ϕ_F) must also be determined to ensure the breaker is as close to a vertical configuration as possible. Such a configuration is desired for the rockbreaking operation since it allows the operator to lower the breaker onto the rock, utilizing the weight of the

[1] Although this algorithm is not outlined in this work, it is based on the trajectory planning algorithms presented within this document in conjunction with the collision detection algorithm found in Chapter 7

rockbreaker to ensure continuous contact and efficient energy transfer between the breaker tip and rock during the breaking operation.

As previously mentioned, trajectory planning is a combination of path planning and time parameterization [62]. In terms of path planning, the proposed method uses a simple point-to-point approach to generate a path between the initial and final rockbreaker poses. Although the rockbreaker's motion between these poses is not critical in the given application, collisions between the rockbreaker and its environment are to be avoided (collision detection-based trajectory validation is implemented in Chapter 7). Given the non-convex nature of the workspace (refer to Figure 3.3), straight-line trajectories between initial and final poses in three-dimensional space are not guaranteed to be feasible. As a result, a decoupling strategy is proposed between the swing joint motion (θ_1) and the motion in the X_2Y_2 plane (θ_2, θ_3 and θ_4) which, for motions constrained to the horizontal work plane, guarantees feasible trajectories so long as $x \geq 0$. Furthermore, due to the high pressure nature of the rockbreaker's hydraulic actuation system, abrupt valve motions are less than desirable seeing as these may lead to unnecessary wear-and-tear of the valve body as well as unwanted vibrations. To ensure smooth valve actuation, the rockbreaker's velocity trajectories are defined using smooth, polynomial-based ramp-up and ramp-down paths at the beginning and end of the planned trajectories.

Once a path has been specified, time parameterization consists of assigning specific time instants to each configuration along the path. This is done here with the objective of minimizing the total rockbreaker transit time between its initial and final poses. This optimization is subjected to constraints aimed at ensuring that flow rate limits through each of the system's proportional valves along with the overall flow rate limit of the hydraulic pump are respected throughout the trajectory. It should be noted that the pressure supplied by the pump is assumed to be sufficient to generate the required flow rates (up to their limits) regardless of the loads felt by the actuators. This is a reasonable assumption given the system is equipped with pressure reduction valves for each actuator and a load-sensing circuit to adjust the hydraulic pump's throughput which compensates for these dynamic loads.

5.2 Calculation of the final pose orientation

Given the user specified final position of the breaker tip (*i.e.* $\mathbf{p}_F$), an optimal breaker orientation as close as possible to a vertical configuration must be found. Initially, the feasibility of a vertical breaker configuration may be verified. This is done by checking that the breaker tip is located in the vertical breaker configuration workspace (Figure 3.3). To do so, the desired final position $\mathbf{p}_F$, combined with an assumed vertical breaker orientation ($\phi_F = 270°$) are set as inputs to the IKP to compute the corresponding revolute joint angles. Moreover, the hydraulic actuator length limits ($\rho_{j_{\min}}$ and $\rho_{j_{\max}}$) are mapped to corresponding revolute joint angle limits ($\theta_{i_{\min}}$ and $\theta_{i_{\max}}$) using Eqs. (3.1) and (3.2). This allows the required revolute joint angles for a vertical breaker orientation to be verified with respect to their ranges of admissible values ($\theta_{i_{\min}} \leq \theta_i \leq \theta_{i_{\max}}$) to ensure their feasibility. If the pose is found to be within the revolute joint limits, the pose is deemed feasible and can be set as the final pose $\mathbf{x}_F = [\mathbf{p}_F^T, \phi_F]^T$, which is then sent to the trajectory planner.

In the event the vertical breaker orientation is found to be infeasible given the desired final position $\mathbf{p}_F$ of the breaker tip, the corresponding orientation of the breaker nearest to the vertical configuration must be found. Given an imposed position of P, the rockbreaker retains one degree of freedom in the X_2Y_2 plane with which the breaker orientation can be modified to bring it toward a vertical orientation until at least one revolute joint limit is reached ($\theta_i = \theta_{i_{\min}}$ or $\theta_i = \theta_{i_{\max}}$ for any i). The optimal breaker orientation should then correspond to a revolute joint limit. To identify the optimal breaker orientation for a given position $\mathbf{p}_F$ of the breaker tip, one must:

1. Compute the position of the breaker tip in the X_2Y_2 plane.
2. Set the k^{th} revolute joint to one of its limits ($\theta_{k_{\min}}$ or $\theta_{k_{\max}}$, $k = 2, 3, 4$).
3. Solve the inverse kinematics of the planar portion of the rockbreaker (θ_2, θ_3 and θ_4 only) for the remaining (unconstrained) revolute joint angles (*i.e.* θ_i with $i \neq k$). This is the equivalent of finding the intersection between two circles.
    a) If the circles do not intersect, skip to step 4.
    b) If the circles intersect, verify the revolute joint angles corresponding to each intersection to ensure they are within their limits of operation ($\theta_{i_{\min}} \leq \theta_i \leq \theta_{i_{\max}}$).

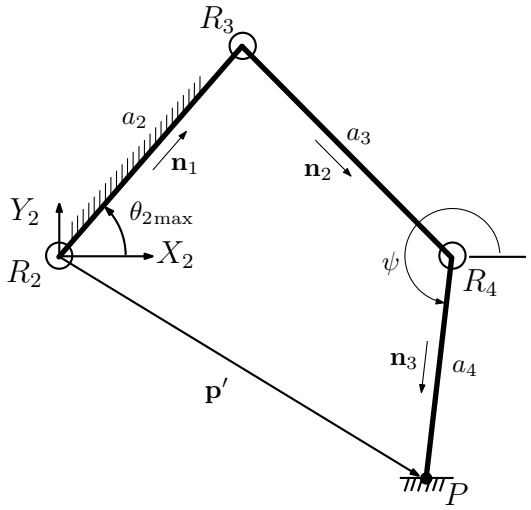


Figure 5.1: Schematic representation of a rockbreaker with its breaker tip located at a prescribed point P with its second revolute joint at its upper limit (*i.e.* $\theta_2 = \theta_{2_{\max}}$).

c) If each revolute joint is within its operating range, add the corresponding breaker orientation to a list of feasible values.

4. Repeat steps 2 and 3 for each limit ($\theta_{k_{\min}}$ and $\theta_{k_{\max}}$) of each joint ($k = 2, 3, 4$). This should generate a maximum of 12 solutions based on the two possible solutions of the planar inverse kinematic problem for each limit of each joint.

5. Identify the optimal breaker orientation (ϕ_F) as the one that is the closest to a vertical configuration from the feasible orientations obtained previously.

Using the aforementioned technique, an optimal breaker orientation can typically be found. If no feasible orientation has been found, this indicates the desired final position of the rockbreaker is outside the rockbreaker's workspace.

The case where $\theta_2 = \theta_{2_{\max}}$ is now used to illustrate this procedure. With θ_2 specified, the position of R_3 in the X_2Y_2 plane is known. Given the corresponding desired position of the breaker tip P, this implies that θ_3 and θ_4 can be adjusted to yield an optimal final breaker orientation ϕ_F. Referring to Figure 5.1, the following vector-loop closure equation may be written:

$$a_3\mathbf{n}_2 = \mathbf{p}' - a_2\mathbf{n}_1 - a_4\mathbf{n}_3 \tag{5.1}$$

where $\mathbf{n}_i$ $(i = 1, 2, 3)$ are unit vectors directed along the rockbreaker's links as shown in Figure 5.1. Furthermore, the position of the breaker tip measured with respect to the origin of the $X_2Y_2Z_2$ frame and expressed within this frame may be obtained using Eq. (3.9). In Eq. (5.1), $\mathbf{p}'$ and $\mathbf{n}_1$ are known based on the specified breaker tip position and the imposed value of θ_2, respectively. Multiplying both sides of Eq. (5.1) by their transpose and rearranging yields:

$$D_1 \cos\psi + D_2 \sin\psi + D_3 = 0 \tag{5.2}$$

where $\mathbf{n}_3 = [\cos\psi, \sin\psi]^T$ and:

$$D_1 = 2a_4(a_2\mathbf{n}_1 - \mathbf{p}')^T\mathbf{i}, \quad D_2 = 2a_4(a_2\mathbf{n}_1 - \mathbf{p}')^T\mathbf{j}, \quad D_3 = a_4^2 - a_3^2 + ||\mathbf{p}' - a_2\mathbf{n}_1||^2 \tag{5.3}$$

Using the tangent of the half-angle substitution with $s = \tan(\psi/2)$, one finally obtains:

$$(D_3 - D_1)s^2 + (2D_2)s + (D_3 + D_1) = 0 \tag{5.4}$$

Solving this quadratic equation yields the following two possible solutions:

$$\psi = 2\tan^{-1} s \tag{5.5}$$

If the discriminant of Eq. (5.4) is negative, the configuration is beyond the rockbreaker's workspace. Otherwise, ϕ_F is obtained from the following equation:

$$\phi_F = \psi - \varphi \tag{5.6}$$

Similar approaches to the one described above are used to identify feasible ϕ_F based on the other joint limits.

5.3 Description of the trajectory planning algorithm

The aim of the trajectory planning algorithm is to generate point-to-point motion between the initial and final rockbreaker poses with minimum transit time while ensuring that flow rate limits throughout are respected. As previously mentioned, trajectories are chosen to be decoupled into two independent but simultaneous motions: swing joint motion and a straight-line motion in the X_2Y_2 plane.

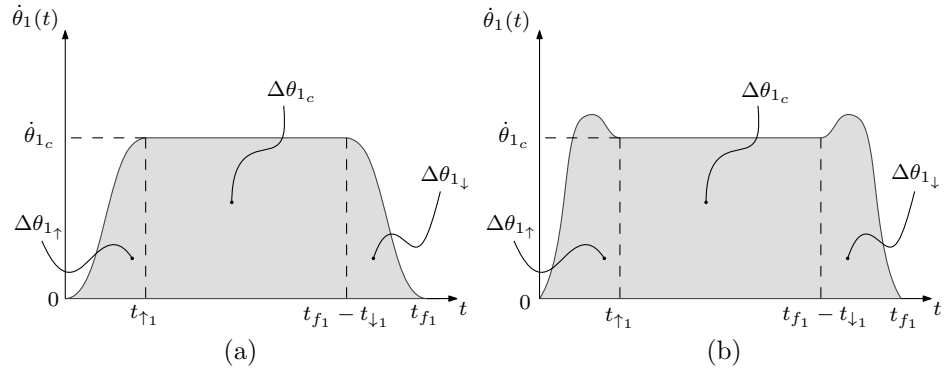


Figure 5.2: Trajectory profile for $\dot{\theta}_1$: (a) Acceptable trajectory profile and (b) undesired trajectory profile due to the presence of a velocity overshoot.

5.3.1 Swing joint trajectory planning

The chosen trajectory profile for the swing joint uses smooth polynomial segments to ramp-up/-down to/from a constant angular velocity $\dot{\theta}_{1_c}$ as illustrated in Figure 5.2(a). The ramp-up and ramp-down segments are chosen to be mirrored copies of each other with respect to the time axis[2]. Referring to Figure 5.2(a), the total angular displacement of the swing joint resulting from the planned trajectory is $\Delta\theta_1 = 2\Delta\theta_{1_\uparrow} + \Delta\theta_{1_c}$ where $\Delta\theta_{1_\uparrow}$ and $\Delta\theta_{1_c}$ are the displacements resulting from the ramp-up/-down segments and from the constant joint velocity segment, respectively. This total displacement is known *a priori* for a desired trajectory as $\Delta\theta_1 = \theta_{1_F} - \theta_{1_I}$ where

$$\theta_{1_I} = \text{atan2}\,(y_I, x_I), \quad \theta_{1_F} = \text{atan2}\,(y_F, x_F) \tag{5.7}$$

Furthermore, still referring to Figure 5.2(a), t_{f_1} defines the total trajectory duration while $t_{\uparrow_1}$ represents the ramp-up/-down duration. The joint displacement resulting from the constant velocity segment is obtained as:

$$\Delta\theta_{1_c} = (t_{f_1} - 2t_{\uparrow_1})\dot{\theta}_{1_c} \tag{5.8}$$

The swing joint trajectory during the ramp-up/down segments is described by the following fifth-order polynomial:

$$\theta_1(t) = h_{0_1} + h_{1_1}t + h_{2_1}t^2 + h_{3_1}t^3 + h_{4_1}t^4 + h_{5_1}t^5, \quad t \in [0, t_{\uparrow_1}] \tag{5.9}$$

[2]The resulting effect being that $t_{\uparrow_1} = t_{\downarrow_1}$ and $\Delta\theta_{1_\uparrow} = \Delta\theta_{1_\downarrow}$

which ensures that the swing joint's angular displacement, velocity and acceleration are smooth and continuous. In order to compute the polynomial's coefficients, the following constraints are applied:

$$\theta_1(0) = \theta_{1_I},\ \theta_1(t_{\uparrow_1}) = \theta_{1_I} + \Delta\theta_{1_\uparrow},\ \dot{\theta}_1(t_{\uparrow_1}) = \dot{\theta}_{1_c},\ \dot{\theta}_1(0) = \ddot{\theta}_1(0) = \ddot{\theta}_1(t_{\uparrow_1}) = 0 \tag{5.10}$$

These constraints can also be formulated in vector-matrix form as:

$$\mathbf{b} = \begin{bmatrix} \theta_1(0) \\ \dot{\theta}_1(0) \\ \ddot{\theta}_1(0) \\ \theta_1(t_{\uparrow_1}) \\ \dot{\theta}_1(t_{\uparrow_1}) \\ \ddot{\theta}_1(t_{\uparrow_1}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & t_{\uparrow_1} & t_{\uparrow_1}^2 & t_{\uparrow_1}^3 & t_{\uparrow_1}^4 & t_{\uparrow_1}^5 \\ 0 & 1 & 2t_{\uparrow_1} & 3t_{\uparrow_1}^2 & 4t_{\uparrow_1}^3 & 5t_{\uparrow_1}^4 \\ 0 & 0 & 2 & 6t_{\uparrow_1} & 12t_{\uparrow_1}^2 & 20t_{\uparrow_1}^3 \end{bmatrix} \begin{bmatrix} h_{0_1} \\ h_{1_1} \\ h_{2_1} \\ h_{3_1} \\ h_{4_1} \\ h_{5_1} \end{bmatrix} = \mathbf{A}\mathbf{h} \tag{5.11}$$

from which the polynomial's coefficients may be obtained as $\mathbf{h} = \mathbf{A}^{-1}\mathbf{b}$. It is noted that $\mathbf{A}$ may easily be verified to be non-singular so long as $t_{\uparrow_1} > 0$. Note that, in order to solve for $\mathbf{h}$, values for $t_{\uparrow_1}$, $\Delta\theta_{1_\uparrow}$ and $\dot{\theta}_{1_c}$ must first be determined. This is done by minimizing the total transit time of the trajectory (*i.e.* t_{f_1}) as follows:

$$\min_{\lambda_1}\ f_1(\boldsymbol{\lambda}_1) = t_{f_1} = \frac{\Delta\theta_1 - 2\Delta\theta_{1_\uparrow}}{\dot{\theta}_{1_c}} + 2t_{\uparrow_1} \tag{5.12}$$

where $\boldsymbol{\lambda}_1 = [t_{\uparrow_1},\ \Delta\theta_{1_\uparrow},\ \dot{\theta}_{1_c}]^T$ is the design vector and where the following constraints are applied:

1. $\Delta\theta_{1_\uparrow} \leq \dfrac{\Delta\theta_1}{2}$
2. $t_{\uparrow_1} \geq 1$ (It should be noted that while a constraint of this nature is deemed necessary to maintain system accelerations at an acceptable level, the minimum value of $t_{\uparrow_1}$ is empirical in nature. If, after testing, the value of $t_{\uparrow_1}$ is deemed unacceptable, it can easily be adjusted to a more suitable value.)
3. $|\dot{\theta}_{1_c}| \leq \dot{\theta}_{1_{\text{lim}}}$ (details to follow)
4. $\text{sign}(\Delta\theta_1)\dddot{\theta}_1(0) \geq 0$ and $\text{sign}(\Delta\theta_1)\dddot{\theta}_1(t_{\uparrow_1}) \leq 0$ (details to follow)

Once optimal values for $t_{\uparrow_1}$, $\Delta\theta_{1_\uparrow}$, $\dot{\theta}_{1_c}$ have been calculated, the coefficients of the ramp-up/-down polynomial may subsequently be computed from Eq. (5.11) such that the swing joint trajectory is

then fully defined. The task that remains is to identify $\dot{\theta}_{1_{\text{lim}}}$ based on the swing joint's flow rate limits.

Constraint 3 (see above list) is based on the need to ensure that the flow rate limit through the swing joint's hydraulic valve (*i.e.* $Q_{1_{\text{max}}}$) is not exceeded throughout the planned trajectory. The required flow rate through the swing joint's valve at any instant is obtained as the first element of $\mathbf{Q}$ in Eq. (3.19), *i.e.*

$$Q_1(t) = A_0|\dot{\rho}_0(t)| + A_1|\dot{\rho}_1(t)| \tag{5.13}$$

where the surface of the pistons (A_0 and A_1) on which the hydraulic fluid is acted upon is based on the direction of the actuator's motion as previously stated in Chapter 3. Since the actuator velocities $\dot{\rho}_0$ and $\dot{\rho}_1$ are dependent on the swing joint's angular velocity $\dot{\theta}_1$, these may be mapped in terms of $\dot{\theta}_1$ from Eqs. (3.10) and (3.12) as $\dot{\rho}_0 = (J_{\beta_0} J_{\rho_0})^{-1}\dot{\theta}_1$ and $\dot{\rho}_1 = (J_{\beta_1} J_{\rho_1})^{-1}\dot{\theta}_1$ respectively. Substituting these results in Eq. (5.13) and solving for $\dot{\theta}_1$, one obtains:

$$\dot{\theta}_1 = \frac{Q_1}{A_0(J_{\beta 0} J_{\rho 0})^{-1} + A_1(J_{\beta 1} J_{\rho 1})^{-1}} \tag{5.14}$$

Setting $Q_1 = Q_{1_{\text{max}}}$ in Eq. (5.14), maximum permissible angular velocities of the swing joint (*i.e.* $\dot{\theta}_{1_{\text{max}}}$) may be obtained for corresponding joint positions (*i.e.* θ_1), the results of which are plotted in Figure 5.3 for the rockbreaker whose parameter values are detailed in Appendix B. From Figure 5.3, it can be observed that there is a quasi-constant region for $\dot{\theta}_{1_{\text{max}}}$ over a wide range of θ_1. By setting $\dot{\theta}_{1_{\text{lim}}} = 0.1066$ rad/s for the purpose of the aforementioned optimization problem constraint, the resulting trajectory is thus guaranteed to remain within the swing joint hydraulic valve's flow rate limit. Clearly, greater angular velocities would be permissible for values of θ_1 close to the boundaries of the joint's range of motion. However, corresponding swing joint configurations suffer from weaker kinematic conditioning where slower rockbreaker motions are advisable.

The previously described constraint, based on the flow rate limit of the swing joint's hydraulic valve, seeks to limit the constant angular velocity portion of the trajectory (*i.e.* where $t_{\uparrow_1} \leq t \leq t_{f_1} - t_{\downarrow_1}$) to a level (*i.e.* $\dot{\theta}_{1_{\text{lim}}}$) that, according to Figure 5.3, is feasible regardless of the joint configuration. Constraint 4, for its part, exists to ensure that $|\dot{\theta}_1(t)| \leq \dot{\theta}_{1_{\text{lim}}}$ is also satisfied during the ramp-up/-down portions of the trajectory. In fact, given the polynomial defined in Eq. (5.9), situations such as the one illustrated in Figure 5.2(b) could occur if no constraint is applied. In order for this

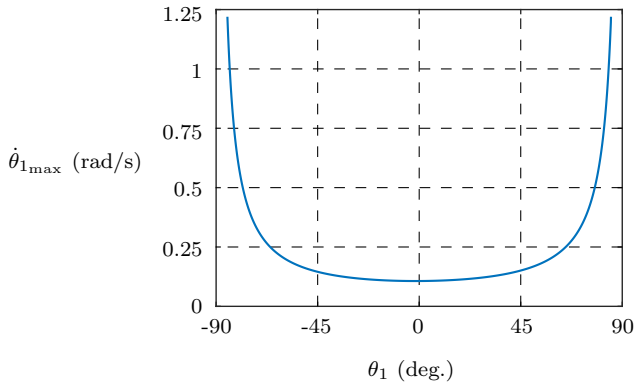


Figure 5.3: Maximum angular velocity of the swing joint (*i.e.* $\dot{\theta}_{1_{\max}}$) based on flow rate limitations in terms of its angular position θ_1.

situation to occur, $\ddot{\theta}_1(t)$ would need to change signs in the time interval $0 \leq t \leq t_{\uparrow_1}$. Combined with the knowledge that $\ddot{\theta}_1(0) = \ddot{\theta}_1(t_{\uparrow_1}) = 0$, this implies that $\dddot{\theta}_1(t) = 0$ would need to occur twice during this same time interval. While this is possible given $\dddot{\theta}_1(t)$ is a 2$^{\text{nd}}$ order polynomial, it would require $\text{sign}(\dddot{\theta}_1(0)) = \text{sign}(\dddot{\theta}_1(t_{\uparrow_1}))$. Thus, by imposing constraint 4 and remembering that the ramp-down segment of the trajectory is a mirror image with respect to time of the ramp-up segment, this situation is avoided and the flow rate limit of the swing joint's hydraulic valve is respected throughout the planned trajectory.

5.3.2 Trajectory planning in the X_2Y_2 plane

With the swing joint's trajectory having been defined, the motion in the X_2Y_2 plane must now be computed. This motion is only dependent on ρ_2, ρ_3 and ρ_4 and is therefore decoupled from the swing motion described in Section 5.3.1. Furthermore, seeing as the overall motion of the rockbreaker is constrained to a horizontal plane (Section 5.1), the motion in the X_2Y_2 plane is in fact constrained to a line parallel to the X_2 axis. The objective of the motion is to move the breaker tip P in a straight line between the initial and final positions expressed in the X_2Y_2 frame ($\mathbf{p}'_I$ and $\mathbf{p}'_F$, respectively) while changing its orientation from ϕ_I to ϕ_F. This may be expressed in terms of a time-dependent function $\sigma(t)$ as follows:

$$\mathbf{p}'(t) = \mathbf{p}'_I + \sigma(t)(\mathbf{p}'_F - \mathbf{p}'_I) \tag{5.15}$$

$$\phi(t) = \phi_I + \sigma(t)(\phi_F - \phi_I) \tag{5.16}$$

where $\sigma \in [0,1]$, $\sigma(0) = 0$, $\sigma(t_{f_2}) = 1$ and t_{f_2} is the total trajectory duration. The objective then becomes to find a suitable trajectory $\sigma(t)$ that minimizes t_{f_2} while ensuring the flow rate limits of the hydraulic valves and pump are not exceeded. The trajectory defined by $\sigma(t)$ is chosen to have, at the velocity level, the same form as the one illustrated in Figure 5.2(a) where θ and σ are interchanged[3]. The ramp-up/-down trajectory segments are thus defined by the following polynomial:

$$\sigma(t) = h_{0_2} + h_{1_2}t + h_{2_2}t^2 + h_{3_2}t^3 + h_{4_2}t^4 + h_{5_2}t^5, \quad t \in [0, t_{\uparrow_2}] \tag{5.17}$$

where the following constraints are applied:

$$\sigma(0) = \dot{\sigma}(0) = \ddot{\sigma}(0) = \ddot{\sigma}(t_{\uparrow_2}) = 0,\ \sigma(t_{\uparrow_2}) = \Delta\sigma_{\uparrow},\ \dot{\sigma}(t_{\uparrow_2}) = \dot{\sigma}_c \tag{5.18}$$

and where $t_{\uparrow_2}$ and $\Delta\sigma_{\uparrow}$ define the duration and the displacement, respectively, of the ramp-up/-down trajectory segments while $\dot{\sigma}_c$ is the velocity of the central trajectory segment. In order to compute the coefficients in Eq. (5.17) in terms of $t_{\uparrow_2}$, $\Delta\sigma_{\uparrow}$ and $\dot{\sigma}_c$, a similar approach to the one introduced in Section 5.3.1 may be used. Prior to that, the design vector $\boldsymbol{\lambda}_2 = [t_{\uparrow_2}, \Delta\sigma_{\uparrow}, \dot{\sigma}_c]^T$ is determined by solving the following optimization problem which seeks to minimize the trajectory duration:

$$\min_{\lambda_2}\ f_2(\boldsymbol{\lambda}_2) = t_{f_2} = \frac{1 - 2\Delta\sigma_{\uparrow}}{\dot{\sigma}_c} + 2t_{\uparrow_2} \tag{5.19}$$

subject to:

1. $\Delta\sigma_{\uparrow} \leq 0.5$ (This ensures the displacement for the velocity ramp-up/-down segments is less than or equal to half of the overall displacement of $\sigma(t)$.)

2. $t_{\uparrow_2} \geq 1$ s (Similar to Section 5.3.1, this constraint may be adjusted empirically during testing.)

3. $\dddot{\sigma}(0) \geq 0$ and $\dddot{\sigma}(t_{\uparrow_2}) \leq 0$ (The basis of this constraint is similar to that of constraint 4 in Section 5.3.1 and is not elaborated upon here.)

4. $\max\limits_{0 \leq t \leq t_{f_2}} [Q_i(t) - Q_{i_{\max}}] \leq 0$ $(i = 2, 3, 4)$ (Ensures that the maximum flow rate for the i^{th} joint's hydraulic valve, *i.e.* $Q_{i_{\max}}$, is not exceeded during the entire trajectory.)

5. $\max\limits_{0 \leq t \leq t_{f_2}} \left[\sum\limits_{i=2}^{4} Q_i(t) \right] + Q_{1_{\max}} - Q_{p_{max}} \leq 0.$ (Ensures that the sum of the flow rates required in each of the hydraulic cylinders never exceeds the pump's maximum flow rate $Q_{p_{\max}}$.)

[3] In this instance, $t_{\uparrow_2} = t_{\downarrow_2}$ and $\Delta\sigma_{\uparrow} = \Delta\sigma_{\downarrow}$

In order to apply constraints 4 and 5 within the optimization, the coefficients in Eq. (5.17) are computed for each candidate design vector $\boldsymbol{\lambda}_2$ such that the $\sigma(t)$ is known for $0 \leq t \leq t_{f_2}$. Using Eqs. (5.15) and (5.16), $\mathbf{p}'(t)$ and $\phi(t)$ can then be computed along with:

$$\dot{\mathbf{p}}'(t) = \dot{\sigma}(t)(\mathbf{p}'_F - \mathbf{p}'_I) \tag{5.20}$$

$$\dot{\phi}(t) = \dot{\sigma}(t)(\phi_F - \phi_I) \tag{5.21}$$

where $\dot{\sigma}(t)$ is obtained from Eq. (5.17). Meanwhile, from Eq. (3.9), the position of the breaker in the $X_1Y_1Z_1$ frame may be computed as:

$$\mathbf{p}(t) = \mathbf{a}_1 + \mathbf{R}_2^1 \mathbf{p}'(t) \tag{5.22}$$

while one also has:

$$\dot{\mathbf{p}}(t) = \mathbf{R}_2^1 \dot{\mathbf{p}}'(t) \tag{5.23}$$

where $\theta_1(t) = \dot{\theta}_1(t) = 0$ has been assumed since only motion in the X_2Y_2 plane is currently being considered. The velocity of the breaker, *i.e.* $\dot{\mathbf{x}}(t) = [\dot{\mathbf{p}}(t)^T, \dot{\phi}(t)]^T$, may then be mapped to corresponding hydraulic actuator speeds by solving Eq. (3.15) for $\dot{\boldsymbol{\rho}}(t)$. The required flow rates to each of the actuators ($j = 2, 3, 4$) are then derived from Eq. (3.19) where, once again, the piston areas subjected to hydraulic pressure are based on the direction of motion for each actuator (as described in Chapter 3). For each design vector candidate $\boldsymbol{\lambda}_2$ that is considered during the optimization process, the required flow rates over the entire trajectory may thus be computed through a discretization of the time axis (*i.e.* over the range $0 \leq t \leq t_{f_2}$) which allows for the verification of constraints 4 and 5. Note that although flow rates may exceed their limits between discretized points, the resulting flow rate remains near its limitation due to the small discretized time step and can be compensated by the controller during the remainder of the motion.

5.3.3 Combination of the swing joint and X_2Y_2 plane trajectories

The result of the trajectory planning algorithms presented in Sections 5.3.1 and 5.3.2 are two decoupled trajectories (for the swing joint and X_2Y_2 planar motion, respectively) of minimized durations t_{f_1} and t_{f_2}. Since both motions may take place simultaneously, and given there is nothing to be gained from having one of the motions be completed prior to the other, the final step in the trajectory planning algorithm is to scale the quicker of the two decoupled trajectories such that the global

(*i.e.* combined) trajectory duration is $t_f = \max(t_{f_1}, t_{f_2})$. In the event $t_{f_1} > t_{f_2}$, the motion within the X_2Y_2 plane will be adjusted as follows:

$$r = \frac{t_{f_1}}{t_{f_2}} \quad \rightarrow \quad t_{\uparrow_2} = t_{\downarrow_2} = rt_{\uparrow_2}, \;\; t_{f_2} = rt_{f_2}, \;\; \dot{\sigma}_c = \frac{\dot{\sigma}_c}{r} \tag{5.24}$$

whereas the swing joint motion will be adjusted as follows if $t_{f_2} > t_{f_1}$:

$$r = \frac{t_{f_2}}{t_{f_1}} \quad \rightarrow \quad t_{\uparrow_1} = t_{\downarrow_1} = rt_{\uparrow_1}, \;\; t_{f_1} = rt_{f_1}, \;\; \dot{\theta}_{1_c} = \frac{\dot{\theta}_{1_c}}{r} \tag{5.25}$$

With these updated values, the trajectory of the scaled motion can be recomputed by identifying the new coefficient values for its ramp-up/-down polynomial (*i.e.* Eq. (5.9) or (5.17)) such that the decoupled trajectories begin and end simultaneously.

5.3.4 Verifying the feasibility of the breaker orientation trajectory

The initial and final poses (*i.e.* $\mathbf{x}_I = [\mathbf{p}_I^T, \phi_I]^T$ and $\mathbf{x}_F = [\mathbf{p}_F^T, \phi_F]^T$) of a desired rockbreaker motion are already verified to be located within its workspace prior to the trajectory generation. As is explained in Section 5.3.2, the initial aim of the trajectory planning approach within the X_2Y_2 plane is to follow a horizontal straight-line path between these two poses in the space defined by $\mathbf{p}'$ and ϕ. It may be observed from Figure 3.3 that all breaker positions $\mathbf{p}'(t)$ defined by Eq. (5.16), once converted using Eq. (3.9) to equivalent positions $\mathbf{p}$ measured with respect to and expressed in the $X_1Y_1Z_1$ frame, are located within the rockbreaker workspace so long as $x \geq 0$. However, these positions may not be reachable at the corresponding breaker orientations defined by Eq. (5.16). The breaker orientation feasibility must therefore be verified throughout the trajectory and, where necessary, $\phi(t)$ as defined by Eq. (5.16) must be adjusted.

In order to verify the feasibility of the breaker orientations described by Eq. (5.16), some of the developments in Section 5.2 can be used. In fact, the search for an optimal ϕ for a given position $\mathbf{p}'$ in Section 5.2 yielded an array of angles corresponding to the various hydraulic actuator limits. It can be inferred that the orientation limits of the breaker for the same position should be found within this array. More specifically, the uppermost and lowermost values within the array may be set as the upper and lower boundaries of ϕ given $\mathbf{p}'$, *i.e.* $\phi_{\max}$ and $\phi_{\min}$, respectively. The validation of the breaker orientation trajectory defined by Eq. (5.16) may then proceed as follows:

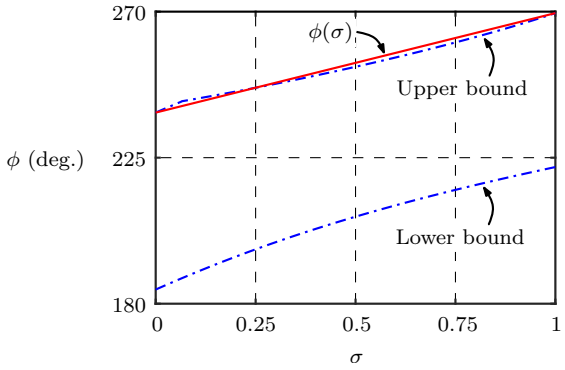


Figure 5.4: Non-feasible linear transition $\phi(\sigma)$ between ϕ_I and ϕ_F.

1. Discretize the trajectory in terms of σ, *i.e.*

$$\mathbf{p}'_k = \mathbf{p}'_I + \left(\frac{k}{N}\right)(\mathbf{p}'_F - \mathbf{p}'_I) \tag{5.26}$$

$$\phi_k = \phi_I + \left(\frac{k}{N}\right)(\phi_F - \phi_I) \tag{5.27}$$

   with $k = 1, 2, \ldots, N-1$ and N is chosen to be sufficiently large (*e.g.* $N = 1000$).

2. For each $\mathbf{p}'_k$, compute the corresponding $\phi_{\min}$ and $\phi_{\max}$ and verify that $\phi_{\min} \leq \phi_k \leq \phi_{\max}$.

The discretization is performed directly in terms of σ (rather than t) as the profile of σ as a function of time has no bearing on the feasibility of the trajectory from the point of view of workspace. Moreover, it is noted that the validation of a discretized trajectory does not provide an absolute guarantee of its feasibility (since not all values of σ are verified) but this approach is considered acceptable in the given context. The validation procedure is illustrated graphically in Figure 5.4 for a sample trajectory with initial and final breaker positions. In Figure 5.4, it may be observed that the initial and final breaker orientations, *i.e.* $\phi_I = \phi(0)$ and $\phi_F = \phi(1)$, are located on the upper limit of the range of admissible orientations which, in this case, is the one closest to the preferred vertical configuration. It may also be seen that the validation of the $\phi(\sigma)$ trajectory would fail in this case since the required breaker orientations do not remain within the feasible region.

The $\phi(\sigma)$ function defining the trajectory must be modified in the event its initial validation fails. It is recalled that ϕ is initially a linear function of σ (refer to Eq. (5.16)). The approach used to modify $\phi(\sigma)$ consists of shifting its "quarter points", *i.e.* where $\sigma = \sigma_q = 0.25q$ with $q = 1, 2, 3$, in order to "guide" the trajectory into a feasible region. For each σ_q, the corresponding $\phi_{\min_q}$ and $\phi_{\max_q}$ are

computed based on $\mathbf{p}'(\sigma_q)$. If $\phi_{\min_q} \leq \phi(\sigma_q) \leq \phi_{\max_q}$ is not satisfied, the value of $\phi(\sigma_q)$ is changed according to:

$$\phi(\sigma_q) = \phi_{\min_q} + \xi||\phi_{\max_q} - \phi_{\min_q}|| \tag{5.28}$$

if $||\phi(\sigma_q) - \phi_{\min_q}|| \leq ||\phi(\sigma_q) - \phi_{\max_q}||$ and

$$\phi(\sigma_q) = \phi_{\max_q} - \xi||\phi_{\max_q} - \phi_{\min_q}|| \tag{5.29}$$

otherwise. In the above equations, $\xi \in [0, 1]$ is an empirically chosen parameter that may be incremented as required until a feasible trajectory is found. Smaller values of ξ are attempted first so that the generated values of $\phi(\sigma_q)$ will satisfy $\phi_{\min_q} \leq \phi(\sigma_q) \leq \phi_{\max_q}$ while remaining close to the boundary that was initially being violated. As such, large deviations from the original $\phi(\sigma)$ trajectory are avoided where possible.

The process by which the $\phi(\sigma)$ may be adjusted is illustrated for a fictitious case in Figure 5.5. In Figure 5.5(a), the original $\phi(\sigma)$ trajectory (*i.e.* based on Eq. (5.16)) is shown along with shaded areas indicating infeasible configurations. The dots along the trajectory represent the so-called "quarter points". The $\phi(\sigma)$ trajectory is clearly not feasible in this case and it must be adjusted by shifting the point for which $\sigma = 0.5$ into the admissible region. Moreover, since the upper bound is the one that is being violated, the adjustment is made using Eq. (5.29) with the result being shown in Figure 5.5(b). At this point, all "quarter points" correspond to feasible configurations and the $\phi(\sigma)$ trajectory consists of a sequence of line segments. In order to simplify the $\phi(\sigma)$ trajectory as much as possible, the algorithm then verifies whether points that were previously unchanged may be shifted to reduce the number of line segments. For the example scenario, this would result in the trajectory shown in Figure 5.5(c). Afterwards, as a final step, corners within the $\phi(\sigma)$ trajectory that are formed by the meeting of two line segments are smoothed using polynomial blends. The width of the blends, *i.e.* the range of σ values centered on the corner location over which the blend is applied, is determined by an empirical parameter $\Delta\sigma$ (a typical value is $\Delta\sigma = 0.025$). The blends are generated using fifth-order polynomials (similar to the ones seen in Section 5.3.1 and 5.3.2) to ensure continuity in terms of displacement, velocity and acceleration. The polynomial's coefficients are computed based on boundary conditions corresponding to:

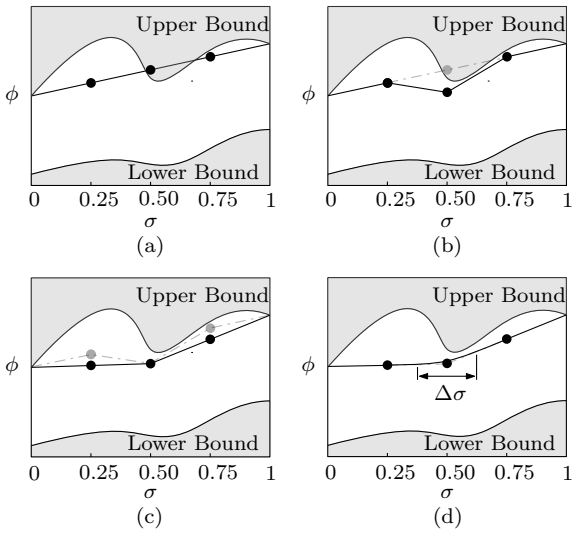


Figure 5.5: Example of an initially non-feasible $\phi(\sigma)$ trajectory that is subjected to quarter point shifts in order to render it feasible.

- The known breaker orientations (based on the two existing line segments) at each end of the blend region.

- The required breaker angular velocities corresponding to the slopes of the two meeting line segments.

- A requirement for zero accelerations at each end of the blend region (given the fact the angular velocity along each line segment is constant).

Once blends have been added to smooth the $\phi(\sigma)$ trajectory, the latter will be piecewise defined as a combination of linear segments and polynomial blends.

After the adjustments to $\phi(\sigma)$ have been completed, the trajectory is discretized into N points and verified for feasibility using Eq. (5.26) along with the newly defined $\phi(\sigma)$ (note that Eq. (5.27) is no longer relevant). If the trajectory remains infeasible, the adjustment process is repeated with an incremented value of ξ. If a feasible trajectory is not found and $\xi = 1$ has been reached, the algorithm is considered to have failed. However, in practice, this was found to occur very rarely and only for scenarios that are highly unlikely to be encountered in a typical rockbreaking operation.

5.4 Results

Using the trajectory planning methodology described in this chapter, a sample trajectory was computed for a rockbreaker whose parameter values are inspired from commercially available systems. The rockbreaker's relevant kinematic prameters along with its hydraulic valve flow rate limits are provided in Table B.1 (Appendix B). Meanwhile, one also has $\varphi = 5.20°$ and $Q_{p_{\max}} = 170$ L/min for this system. Lastly, the physical properties of the system's hydraulic actuators are listed in Table B.2.

In order to illustrate the trajectory planning methodology, a trajectory bringing the breaker tip from an initial position $\mathbf{p}_I = [5.5, 4.8, -0.5]^T$ m to a final position $\mathbf{p}_F = [3, -3, -0.5]^T$ m is generated. These positions describe a motion that begins with the breaker tip positioned at the outermost left corner of the work plane and ends when the breaker tip reaches a position at the innermost right corner of the work plane (this can best be visualized in Figure 3.1). It is noted that, for the duration of the trajectory, the elevation of the rockbreaker tip is constrained to remain constant. The optimal breaker orientations (ϕ_I and ϕ_F) corresponding to the initial and final trajectory positions must now be found. It is possible to perform a visual check to see if a vertical breaker orientation is feasible or not for a given position based on Figure 3.3. With this in mind, it is useful to point out that Figure 3.3 was prepared specifically for the sample rockbreaker whose kinematic parameter values have been provided in this section. Given that Figure 3.3 corresponds to the case where $\theta_1 = 0$, the breaker positions are first mapped to coordinates $(\mathbf{p}^T\mathbf{i}_2, z)$ which may then be referred to the abscissa and ordinate of Figure 3.3 (note that $\mathbf{i}_2$ is a unit vector along the X_2 axis). For the cases of $\mathbf{p}_I$ and $\mathbf{p}_F$, this yields $(7.3, -0.5)$ m and $(4.24, -0.5)$ m, respectively. From Figure 3.3, it can be seen that $\mathbf{p}_F$ is located within the vertical breaker coverage area and therefore $\phi_F = 270°$. However, the same cannot be said for $\mathbf{p}_I$. Using the method described in Section 5.2, the optimal breaker orientation at $\mathbf{p}_I$ is found to be $\phi_I = 312.74°$. From $\mathbf{p}_I$ and $\mathbf{p}_F$, the initial and final swing joint angles are computed using Eq. (5.7) as $\theta_{1_I} = 41.11°$ and $\theta_{1_F} = -45°$, respectively. The resulting trajectory using the algorithm from Section 5.3.1 yields the angular displacement and velocity trajectories shown in Figure 5.6. It should be noted from Figure 5.6(b) that $\dot{\theta}_1(t)$ respects the constraint that was imposed based on the joint's flow rate limit. Lastly, the total trajectory time is found to be $t_{f_1} = 15.04$ s. Using Eq. (3.9), $\mathbf{p}_I$ and $\mathbf{p}_F$ can be described relative to the X_2Y_2 plane

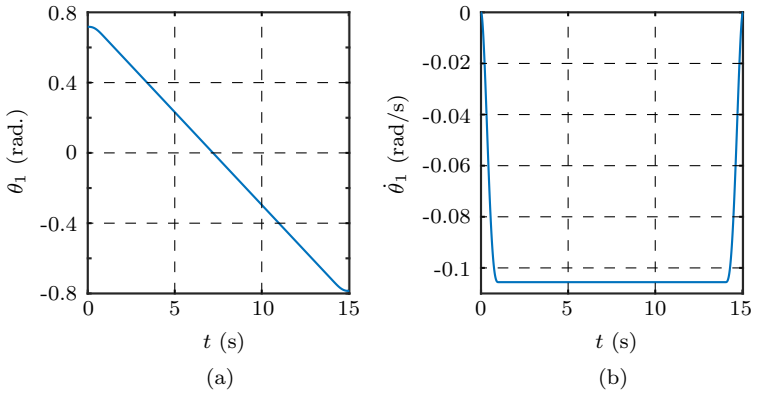


Figure 5.6: Optimized swing joint trajectory between $\mathbf{p}_I$ and $\mathbf{p}_F$: (a) $\theta_1(t)$ and (b) $\dot{\theta}_1(t)$.

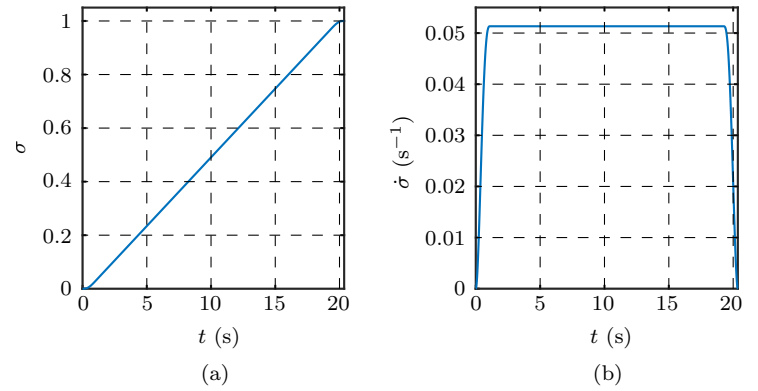


Figure 5.7: Optimized X_2Y_2 planar trajectory from $(\mathbf{p}'_I, \phi_I)$ to $(\mathbf{p}'_F, \phi_F)$ in terms of the $\sigma(t)$ function: (a) $\sigma(t)$ and (b) $\dot{\sigma}(t)$.

as $\mathbf{p}'_I = [6.93, -1.60, 0]^T$ m and $\mathbf{p}'_F = [3.88, -1.60, 0]^T$ m, respectively. Resulting trajectories for $\sigma(t)$ and $\dot{\sigma}(t)$ are shown in Figure 5.7. Note that, in this case, the resulting trajectory for $\phi(t)$ as described by Eq. (5.16) is feasible (*i.e.* it does not need to be adjusted according to the methods presented in Section 5.3.4) and the total trajectory time for the planar motion is computed as $t_{f_2} = 20.36$ s. From the computed total time for each trajectory, it is observed that the swing trajectory is shorter in duration than the planar trajectory ($t_{f_1} < t_{f_2}$). To ensure these motions occur simultaneously, the swing trajectory is adjusted using Eq. (5.25). Afterwards, the swing joint and X_2Y_2 plane trajectories may be combined to yield the Cartesian trajectories seen in Figure 5.8 (note that $z = -0.5$ m is constant throughout the entire motion and is not plotted). Based on the solution to the rockbreaker's inverse kinematic problem (Chapter 3), the trajectory may be mapped to corresponding hydraulic actuator lengths which are plotted in Figure 5.9. Moreover, using Eq. (3.15), the hydraulic actuator velocities (*i.e.* $\dot{\rho}_j$) may be computed for the entire trajectory and then converted to corresponding

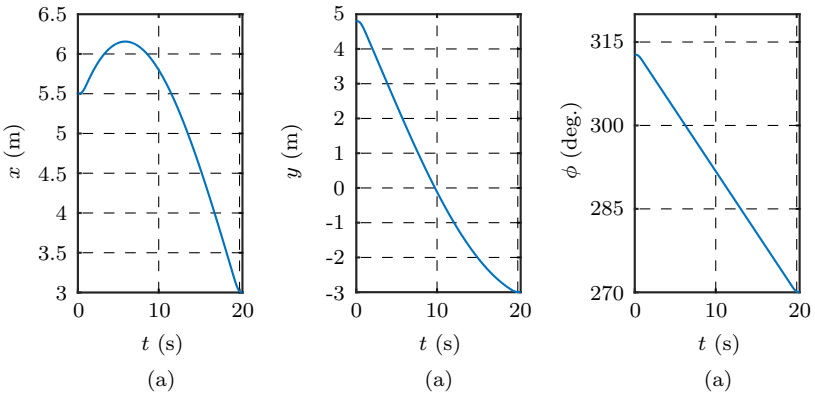


Figure 5.8: Optimized Cartesian trajectory between position $\mathbf{p}_I$ and $\mathbf{p}_F$: (a) $x(t)$, (b) $y(t)$ and (c) $\phi(t)$.

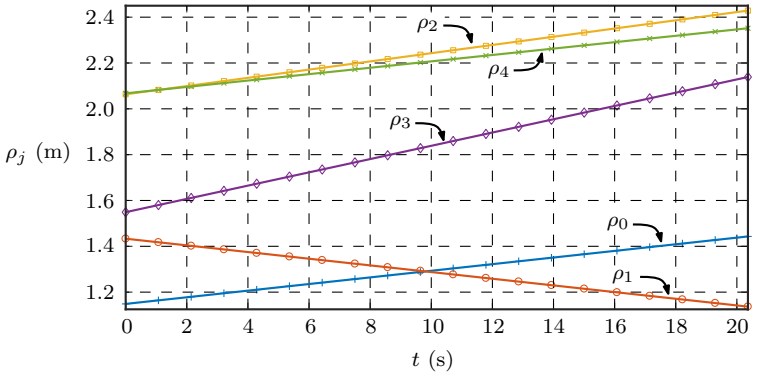


Figure 5.9: Fluctuations on the hydraulic actuator lengths throughout the optimized trajectory.

hydraulic valve flow rates (*i.e.* Q_i) from Eq. (3.19). In order to better gauge the trajectory planning algorithm's efficiency in terms of maximizing the use of the rockbreaker's capacities with regards to flow rate limits, normalized flow rates with respect to time $\hat{Q}_i(t) = Q_i(t)/Q_{i_{\max}}$ are plotted in Figure 5.10. It may be noted that at least one of Q_2, Q_3 or Q_4 is expected to reach its limit during the motion according the constraints introduced in Section 5.3.2. In the specific case considered here, one may observe that $Q_3 = Q_{3_{\max}}$ when $t \approx 1$ s. Meanwhile, the swing joint's flow rate never approaches its limit since the swing joint trajectory time was scaled upwards to coincide with the greater planar trajectory duration.

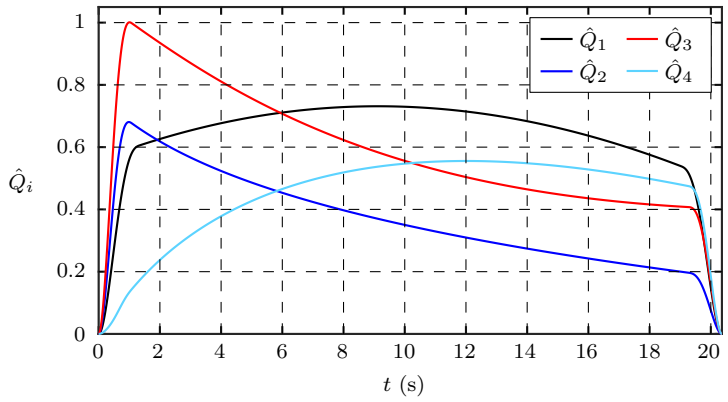


Figure 5.10: Normalized flow rates in the rockbreaker's proportional valves throughout the optimized trajectory.

5.5 Conclusion

A trajectory planning algorithm for the semi-automation of hydraulic rockbreaker machines has been presented in this chapter. The proposed algorithm generates a trajectory bringing the breaker to a user-specified position in minimal time while not exceeding flow rate limit constraints due to the hydraulic pump and servovalves. By decoupling the rockbreaker motion into a rotation about its vertical swing joint combined with a motion constrained to a vertical plane perpendicular to the remaining joints, trajectories are ensured to be located within the machine's workspace. Meanwhile, smooth and continuous fluctuations in the rockbreaker's position, velocity and acceleration are ensured by the use of polynomial-based transitions within the planned trajectories.

Referring to the plot of the rockbreaker's normalized flow rates in Figure 5.10, one may observe that the flow rate limits are only reached at specific points along the trajectory (in this case when $t \approx 1$ seconds). When compared to human operators, who typically actuate the valves to their full capacities during large portions of rockbreaker motions, the resulting trajectories may not be time optimal. As such, a method by which the trajectory planning methodology may fully exploit the rockbreaker's flow rate capacities so as to better approximate the transit times achieved by human operators while ensuring the smooth and predictable operation of the rockbreaker should be developed (and is presented in Chapter 6). Such an approach would be very similar to the one proposed in [62] and implemented in an efficient algorithm in [67]. However, the method described in those works does not guarantee continuous accelerations throughout the resulting trajectories which

is an issue given the nature of a rockbreaker's hydraulic actuation system.

Chapter 6

Trajectory planning using continuous flow rate scaling

The trajectory planning algorithm introduced in Chapter 5 yielded time parameterized trajectories, scaled using a constant scaling factor, which ensured flow rate limitations were respected. A similar approach was also used in [19], however, this resulting strategy severely hampers efforts to minimize trajectory duration. To ensure time-optimal path parameterization, one may utilize the approach described in [62] for which an efficient implementation is proposed in [67], although it has apparently not yet been adapted to the specific case of systems subjected to flow rate limits. Moreover, the method does not guarantee trajectories which are continuous at the acceleration level which, while acceptable in some cases, is considered a notable limitation for hydraulically-driven systems. This chapter provides insight on the development of improvements to the algorithm described in Chapter 5 to render trajectories time optimal for given paths. This is accomplished by the algorithm by using a continuously variable scaling of flow rates throughout the trajectory to ensure that, with the exception of the latter's acceleration and deceleration phases, at least one hydraulic actuator is operating at its maximum flow rate limit at all times. This more closely replicates the human operation of rockbreakers in terms of exploiting the capacities of the latter. However, contrary to the motions generated by human operators, the method proposed in this chapter reduces wear and tear on the rockbreaker's components by imposing smooth motions. The proposed method naturally leads to a decrease in the duration of planned trajectories.

The remainder of this chapter is organized as follows. The proposed trajectory planning methodology is detailed in Section 6.1, followed by a discussion on pose identification in Section 6.2. Parameterization of the position and orientation trajectories is then provided in Sections 6.3, after which, Sections 6.4 and 6.5 describe the two-phase operation which ensures optimized trajectory generation. After an optimized trajectory is generated, it must undergo post processing, which is discussed in Section 6.6. The application of the algorithm is then illustrated through sample case studies in Section 6.7 and finally, Section 6.8 summarizes the chapter's contributions, discusses their limitations and presents some ideas for future work.

6.1 Overview of the proposed trajectory planning methodology

The operation of a semi-automated rockbreaker, as was initially defined in Chapter 1 may be summarized as follows:

1. Specification by the human operator, through a graphical user interface (GUI) equipped with a live video feed of the work area, of the location on the grizzly of a boulder needing to be fragmented.

2. Automated motion of the rockbreaker to a position located above the identified boulder with an optimized orientation of its breaker.

3. Fragmentation of the boulder by the human operator through the use of a teleoperation interface.

The automated motion is assumed to take place with the chisel tip remaining within a work plane (see Figure 3.1) that is located above the grizzly at an empirically-determined height that is greater than that of the largest boulder. However, the motion of the rockbreaker from its initial location (*i.e.* where it was left following the fragmentation of a boulder through teleoperation) to a location on the work plane may also be automated. The role of trajectory planning within the semi-automated rockbreaker system is to produce the required time series of actuator displacements, velocities and acceleration that are subsequently used as the inputs to a motion control algorithm.

With productivity and cost effectiveness in mind, the proposed trajectory planning approach aims to minimize the duration of rockbreaker trajectories subject to the flow rate limits of its proportional valves and hydraulic pump. While the targeted application is principally concerned with moving the rockbreaker to a desired location, this must be done while following a known trajectory such that collisions between the rockbreaker and its surrounding environment may be avoided. With this in mind, the following two types of trajectories may be considered:

- **Decoupled swing joint motion approach:** In this case the motion is decoupled into a combination of a swing joint (*i.e.* θ_1) motion and a horizontal straight-line motion in the X_2Y_2 plane. This approach has the benefit of ensuring that trajectories are located within the rockbreaker's non-convex workspace so long as the initial and final poses are both in the workspace but outside the shaded region in Figure 3.3 that is identified as practically unreachable (this region is theoretically reachable by the rockbreaker but that would require it to pass through the ground beneath its own pedestal which is not feasible).

- **Cartesian straight-line trajectory approach:** Due to the rockbreaker's workspace not being convex (see Figure 3.3), the feasibility of straight-line trajectories between two arbitrary chisel tip positions cannot be guaranteed. However, such trajectories (once their feasibility has been verified) are useful in situations where the decoupled swing joint motion approach leads to collisions between the rockbreaker and its surrounding environment as well as when the rockbreaker must initially be moved to the work plane (*i.e.* prior to transiting to the targeted boulder position).

The work presented in Chapter 5 focused on the use of the decoupled swing joint motion approach where $\sigma(t)$ was defined as a parameter representing motion along a straight line in the X_2Y_2 plane. The trajectory at the velocity level (*i.e.* $\dot{\theta}_1(t)$ or $\dot{\sigma}(t)$) was assumed to have the form illustrated in Figure 6.1, consisting of gradual ramp-up/-down trajectory segments that transition the rockbreaker between rest and constant velocity motion. The trajectory duration was minimized subject to the limits of the proportional valve and hydraulic pump flow rates. This was done in such a way that the flow rate capacity would be fully exploited (*i.e.* in a given configuration the rockbreaker is moving at maximum speed in the desired direction) at least once during the trajectory which, given the

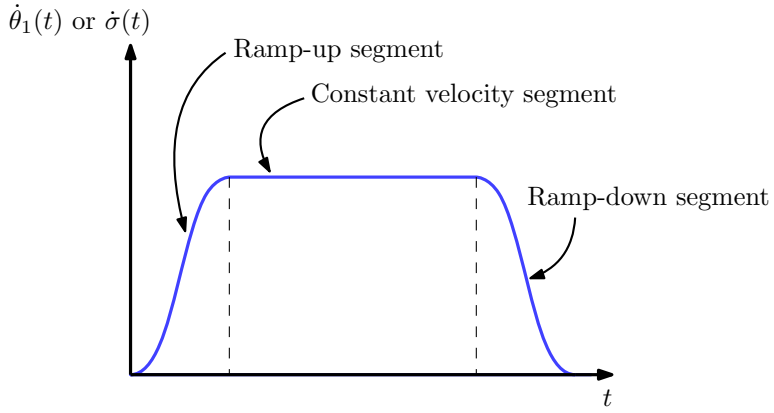


Figure 6.1: Assumed form of the velocity-level trajectories generated using method presented in Chapter 5.

imposed constant velocity segment, is the best that can be done.

The method proposed in this chapter seeks to improve on the approach described in Chapter 5 by ensuring that the rockbreaker's flow rate capacity is fully exploited throughout most of its trajectory (the exception being the ramp-up/-down segments). It is believed that this more closely replicates the human operation of a rockbreaker while providing the added benefits of following a prescribed trajectory while ensuring smooth and continuous motion to reduce wear and tear on the machine. In what follows, the method will be described and illustrated for the case of straight-line trajectories in Cartesian space. However, it could also easily be generalized to the case of decoupled swing joint motion.

6.2 Identification of the initial and final trajectory poses

The proposed methodology is focused on generating trajectories that bring the rockbreaker chisel tip from an initial position $\mathbf{p}_I$ to a final position $\mathbf{p}_F$ along a straight line in Cartesian space. At the same time, the breaker's orientation must transition between initial and final angles ϕ_I and ϕ_F, respectively. The rockbreaker's initial pose (*i.e.* $\mathbf{p}_I$ and ϕ_I) is obtained from the solution of the DKP based on data acquired from joint sensors. It will henceforth be assumed that the rockbreaker's chisel tip is initially located in the work plane (refer to Figure 3.1). However, preliminary trajectories moving the rockbreaker to the work plane may also be planned using the proposed method (details are not provided for the sake of brevity). The X_1Y_1 plane projection of the desired final position

of the rockbreaker is specified by the user through the GUI based on the location on the grizzly of the boulder to be fragmented. Combined with the knowledge of the work plane's elevation, this leads to the desired final position $\mathbf{p}_F$. With its chisel tip in a given position, the rockbreaker retains the ability to modify its breaker orientation. Generally speaking, a vertical breaker orientation (*i.e.* $\phi = 270$ degrees) is preferred as it allows for the leveraging of the rockbreaker's weight to fragment boulders. However, as observed in Figure 3.3, vertical breaker orientations are not always feasible within the rockbreaker workspace. In order to determine the final breaker orientation, the feasibility of a vertical breaker orientation for a given $\mathbf{p}_F$ is first verified through the solution of the IKP. If the resulting hydraulic actuator lengths are found to be within their respective operating ranges (*i.e.* $\rho_{j_{\min}} \leq \rho_j \leq \rho_{j_{\max}}$), the final breaker orientation is set to $\phi_F = 270$ degrees. Otherwise, the optimal final breaker orientation (*i.e.* the one closest to a vertical orientation) may be determined using the procedure outlined in Section 5.2, which is based on the fact that the optimal orientation corresponds to a case where at least one hydraulic actuator is at a limit of its operating range.

6.3 Parameterization of the position and orientation trajectories

The straight-line trajectory that is to be followed by the rockbreaker's chisel in Cartesian space from an initial position $\mathbf{p}_I$ to a final position $\mathbf{p}_F$ may be expressed as

$$\mathbf{p}(\sigma) = \mathbf{p}_I + \sigma \cdot (\mathbf{p}_F - \mathbf{p}_I) \tag{6.1}$$

where σ is a monotonically increasing parameter with $\sigma \in [0, 1]$. As previously mentioned, the location of $\mathbf{p}_I$ and $\mathbf{p}_F$ within the rockbreaker workspace is a necessary but insufficient condition for the trajectory described by Eq. (6.1) to be feasible due to the non-convexity of the workspace. In the event this trajectory is not feasible, the semi-automated rockbreaker system would resort to a trajectory planned with the decoupled swing joint approach. Alternatively, the planning of straight-line trajectories using via points to remain within the rockbreaker workspace could also be considered.

As far as the breaker orientation is concerned, its transition from ϕ_I to ϕ_F is also initially assumed to occur linearly such that

$$\phi(\sigma) = \phi_I + \sigma \cdot (\phi_F - \phi_I) \tag{6.2}$$

However, for a given rockbreaker position $\mathbf{p}(\sigma)$ based on Eq. (6.1), the feasibility of the corresponding $\phi(\sigma)$ as described by Eq. (6.2) is not guaranteed. This may be verified by solving the IKP for $\mathbf{p}(\sigma)$ and $\phi(\sigma)$ and checking to see if the resulting hydraulic actuator lengths are within their respective operating ranges. If this is not the case, Eq. (6.2) is modified empirically according to a method that has previously been described in Section 5.3.4. The result is a piecewise definition of the function $\phi(\sigma)$ as of a combination of linear segments and 3rd-order polynomial blend segments. In the remainder of the chapter, the knowledge of the $\phi(\sigma)$ function is assumed as is the existence of its first order derivative, *i.e.* $d\phi/d\sigma$ (the method presented in Section 5.3.4 ensures the latter).

Given the Cartesian path described as $\mathbf{p}(\sigma)$ and $\phi(\sigma)$, the remaining task is the time parameterization of this path, defined by $\sigma(t)$. This is done with the goal of minimizing the trajectory duration while ensuring smooth and continuous changes in the rockbreaker's velocity. Moreover, as previously noted, the optimization is constrained by the flow rate limits associated with the rockbreaker's proportional valves and hydraulic pump. In fact, with the rockbreaker in a given configuration and with the direction of its velocity known, its speed will be constrained by its flow rate limits. The approach that is adopted here to meet these objectives consists of a two-phase process, whose details are described in Sections 6.4 and 6.5.

6.4 Trajectory optimization: Phase I

In the first phase of the trajectory optimization, a trajectory profile for $\sigma(t)$ that ensures smooth and continuous velocity variations is adopted and its parameters are optimized with the minimization of the trajectory duration in mind.

6.4.1 Description of the initial trajectory profile

The role of the initial trajectory profile assigned to $\sigma(t)$ is to ensure smooth and continuous variations in the rockbreaker's velocity. As such, the profile is more easily described at the velocity level. The adopted profile is illustrated in Figure 6.2. The trajectory duration is t_f such that $\sigma(0) = 0$ and $\sigma(t_f) = 1$. Moreover, the fact that the rockbreaker is at rest in both its initial and final poses implies that $\dot{\sigma}(0) = \dot{\sigma}(t_f) = 0$. The first segment of the profile is a smooth polynomial ramp-up of $\dot{\sigma}(t)$ from

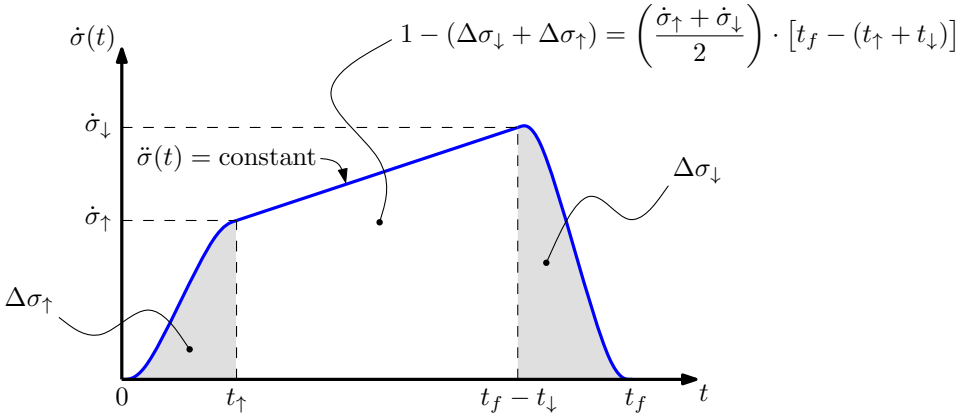


Figure 6.2: Initial trajectory profile at the velocity level.

0 to $\dot{\sigma}_\uparrow$. This ramp-up occurs over a time period $t_\uparrow$ and engenders a corresponding change $\Delta\sigma_\uparrow$ of $\sigma(t)$. Likewise, a smooth polynomial ramp-down segment of duration $t_\downarrow$ transitions $\dot{\sigma}(t)$ from $\dot{\sigma}_\downarrow$ to 0 with a corresponding decrease $\Delta\sigma_\downarrow$ of $\sigma(t)$. The ramp-up and ramp-down trajectory segments are connected by a central segment where $\dot{\sigma}(t)$ transitions from $\dot{\sigma}_\uparrow$ to $\dot{\sigma}_\downarrow$ with constant $\ddot{\sigma}(t)$.

In order to ensure continuity up to the acceleration level, the ramp-up and ramp-down segments are defined as 5th-order polynomials at the displacement level[1], *i.e.*

$$\sigma(t)=\sum_{p=0}^{5}h_{\uparrow,p}\cdot t^p \quad , \quad 0\le t\le t_\uparrow \tag{6.3}$$

$$\sigma(t)=\sum_{p=0}^{5}h_{\downarrow,p}\cdot t^p \quad , \quad t_f-t_\downarrow\le t\le t_f \tag{6.4}$$

where $h_{\uparrow,p}$ and $h_{\downarrow,p}$ are coefficients that may be determined from a set of boundary conditions. The boundary conditions for the ramp-up segment are

$$\sigma(0)=0,\ \sigma(t_\uparrow)=\Delta\sigma_\uparrow,\ \dot{\sigma}(0)=0,\ \dot{\sigma}(t_\uparrow)=\dot{\sigma}_\uparrow,\ \ddot{\sigma}(0)=0,\ \ddot{\sigma}(t_\uparrow)=\frac{\dot{\sigma}_\downarrow-\dot{\sigma}_\uparrow}{t_f-(t_\uparrow+t_\downarrow)} \tag{6.5}$$

while for the ramp-down segment one has

$$\sigma(t_f-t_\downarrow)=1-\Delta\sigma_\downarrow,\ \sigma(t_f)=1,\ \dot{\sigma}(t_f-t_\downarrow)=\dot{\sigma}_\downarrow,\ \dot{\sigma}(t_f)=0,\ \ddot{\sigma}(t_f-t_\downarrow)=\frac{\dot{\sigma}_\downarrow-\dot{\sigma}_\uparrow}{t_f-(t_\uparrow+t_\downarrow)},\ \ddot{\sigma}(t_f)=0 \tag{6.6}$$

From the boundary conditions, solving for the coefficients of the polynomials in Eqs. (6.3) and (6.4) amounts to the solution of two linear systems of equations. As far as the central trajectory segment

[1] It would be straightforward to use 7th-order polynomials instead to ensure the acceleration is both continuous and smooth if this was desired.

is concerned, it is expressed at the displacement level as

$$\sigma(t) = \Delta\sigma_{\uparrow} + \dot{\sigma}_{\uparrow}(t - t_{\uparrow}) + \frac{1}{2}\left[\frac{\dot{\sigma}_{\downarrow} - \dot{\sigma}_{\uparrow}}{t_f - (t_{\uparrow} + t_{\downarrow})}\right](t - t_{\uparrow})^2 \quad , \quad t_{\uparrow} \leq t \leq t_f - t_{\downarrow} \tag{6.7}$$

Referring to Figure 6.2, the trajectory duration may be expressed as

$$t_f = t_{\uparrow} + t_{\downarrow} + \frac{2\big[1 - (\Delta\sigma_{\uparrow} + \Delta\sigma_{\downarrow})\big]}{\dot{\sigma}_{\uparrow} + \dot{\sigma}_{\downarrow}} \tag{6.8}$$

It may thus be observed from Eqs. (6.3) through (6.8) that the trajectory profile illustrated in Figure 6.2 is completely defined by $t_{\uparrow}$, $t_{\downarrow}$, $\Delta\sigma_{\uparrow}$, $\Delta\sigma_{\downarrow}$, $\dot{\sigma}_{\uparrow}$ and $\dot{\sigma}_{\downarrow}$.

6.4.2 Constraints on $\dot{\sigma}_{\uparrow}$ and $\dot{\sigma}_{\downarrow}$

It is quite natural to associate the minimization of the trajectory duration with the full exploitation of the rockbreaker's flow rate capacity. However, it is clear that this capacity cannot be fully exploited during the ramp-up and ramp-down trajectory segments since smooth velocity variations (to and from rest) must be ensured. Instead, constraints are imposed whereby at least one of the flow rate limits (related to one of the proportional valves or to the hydraulic pump) is reached at each of $t_{\uparrow}$ and $t_f - t_{\downarrow}$. This produces a trajectory profile in a form appropriate for the second phase of the trajectory optimization where the available flow rate capacity is fully exploited throughout the profile's central segment (*i.e.* between $t_{\uparrow}$ and $t_f - t_{\downarrow}$). The detailed explanation of the constraints' application will be provided only as it relates to the ramp-up segment of the trajectory profile. However, these explanations naturally extend to the ramp-down segment.

The constraint on $\dot{\sigma}_{\uparrow}$ aims to ensure that at least one flow rate limit is reached at the end of the ramp-up segment where $\sigma(t_{\uparrow}) = \Delta\sigma_{\uparrow}$. The rockbreaker's corresponding position and orientation may be found as $\mathbf{p}_{\uparrow} = \mathbf{p}(\Delta\sigma_{\uparrow})$ and $\phi_{\uparrow} = \phi(\Delta\sigma_{\uparrow})$ (refer to Section 6.3). The velocity of the rockbreaker's chisel tip in the same configuration is found by taking the time derivative of Eq. (6.1), *i.e.*

$$\dot{\mathbf{p}}_{\uparrow} = \dot{\sigma}_{\uparrow} \cdot (\mathbf{p}_F - \mathbf{p}_I) \tag{6.9}$$

The breaker's angular velocity, for its part, may be obtained as

$$\dot{\phi}_{\uparrow} = \frac{d}{dt}\Big\{\phi\big[\sigma(t)\big]\Big\}\bigg|_{\sigma=\Delta\sigma_{\uparrow}} = \left(\frac{d\phi}{d\sigma} \cdot \frac{d\sigma}{dt}\right)\bigg|_{\sigma=\Delta\sigma_{\uparrow}} = \left(\frac{d\phi}{d\sigma}\right)\bigg|_{\sigma=\Delta\sigma_{\uparrow}} \cdot \dot{\sigma}_{\uparrow} \tag{6.10}$$

where $d\phi/d\sigma = \phi_F - \phi_I$ if Eq. (6.2) is in effect. Otherwise, if adjustments to the $\phi(\sigma)$ function were required due to infeasible breaker orientations along the trajectory (refer to Section 6.3) the derivative of $d\phi/d\sigma$ would need to be evaluated based on the piecewise definition of $\phi(\sigma)$ in terms of linear and 3rd-order polynomial blend segments. The translational and angular velocities may then be combined to yield the rockbreaker's Cartesian velocity vector $\dot{\mathbf{x}}_\uparrow = \dot{\sigma}_\uparrow \mathbf{e}$ where

$$\mathbf{e} = \left[(\mathbf{p}_F - \mathbf{p}_I)^T \ , \ \left(\frac{d\phi}{d\sigma} \right) \Bigg|_{\sigma=\Delta\sigma_\uparrow} \right]^T \tag{6.11}$$

Based on Eq. (3.15), the vector of actuator velocities corresponding to $\dot{\mathbf{x}}_\uparrow$ may be obtained as $\dot{\boldsymbol{\rho}}_\uparrow = \mathbf{J}\dot{\mathbf{x}}_\uparrow = \mathbf{J}\mathbf{e}\dot{\sigma}_\uparrow$ where $\dot{\sigma}_\uparrow = 1$ is set arbitrarily for the moment. Referring to Eq. (3.19), the vector of flow rates at the end of the ramp-up segment are then computed as

$$\mathbf{Q}_\uparrow = \mathbf{J}_q |\dot{\boldsymbol{\rho}}_\uparrow| \tag{6.12}$$

Defining a normalized flow rate vector at $t_\uparrow$ as

$$\hat{\mathbf{Q}}_\uparrow = \left[\frac{Q_{1_\uparrow}}{Q_{1_{\max}}} \ , \ \frac{Q_{2_\uparrow}}{Q_{2_{\max}}} \ , \ \frac{Q_{3_\uparrow}}{Q_{3_{\max}}} \ , \ \frac{Q_{4_\uparrow}}{Q_{4_{\max}}} \ , \ \frac{Q_{p_\uparrow}}{Q_{p_{\max}}} \right]^T \tag{6.13}$$

and its k^{th} element as $\hat{Q}_k$ $(k = 1, 2, \ldots, 5)$, a vector of optimal flow rates (where at least one of the proportional valves or the hydraulic pump is providing its maximum flow rate) may be obtained as

$$\tilde{\mathbf{Q}}_\uparrow = \frac{\mathbf{Q}_\uparrow}{\max\limits_k (\hat{Q}_{k_\uparrow})} \tag{6.14}$$

Referring to Eq. (6.12) where $\mathbf{Q}_\uparrow = \tilde{\mathbf{Q}}_\uparrow$ is now substituted, one obtains the optimal $\dot{\sigma}_\uparrow$ for given $\sigma(t_\uparrow) = \Delta\sigma_\uparrow$ as

$$\dot{\sigma}_\uparrow = \frac{1}{\max\limits_k (\hat{Q}_{k_\uparrow})} \tag{6.15}$$

In other words, $\dot{\sigma}_\uparrow$ has now been expressed in terms of $\Delta\sigma_\uparrow$ in a way which ensures that the trajectory profile illustrated in Figure 6.2 will fully exploit the rockbreaker's flow rate capacities at $t_\uparrow$. Similar developments would allow $\dot{\sigma}_\downarrow$ to be expressed in terms of $\Delta\sigma_\downarrow$ so that flow rate capacities are also fully exploited at $t_f - t_\downarrow$. Effectively, this means that the trajectory profile in Figure 6.2 is now defined by only four parameters: $t_\uparrow$, $t_\downarrow$, $\Delta\sigma_\uparrow$ and $\Delta\sigma_\downarrow$.

6.4.3 Constrained minimization of the trajectory duration

The final step in the first phase of the trajectory optimization methodology consists of adjusting the trajectory profile illustrated in Figure 6.2 so that the total trajectory duration t_f is minimized. Referring to Eq. (6.8) and taking into account the final observation of Section 6.4.2, it may be seen that t_f is a function of $t_\uparrow$, $t_\downarrow$, $\Delta\sigma_\uparrow$ and $\Delta\sigma_\downarrow$. The trajectory duration t_f could theoretically be minimized over the search space defined by these parameters. However, in order to maintain rockbreaker accelerations at acceptable levels, the ramp-up/-down segment durations are empirically chosen prior to the optimization, where longer durations have a direct impact on limiting rockbreaker accelerations (typically $t_\uparrow, t_\downarrow \geq 1$ second). The optimization problem may thus be formulated as the minimization of $t_f = f(\Delta\sigma_\uparrow, \Delta\sigma_\downarrow)$ subject to the following constraints

$$g_1 = \dddot{\sigma}(0) \geq 0, \quad g_2 = \dddot{\sigma}(t_\uparrow) \leq 0, \quad g_3 = \dddot{\sigma}(t_f - t_\downarrow) \leq 0, \quad g_4 = \dddot{\sigma}(t_f) \geq 0 \tag{6.16}$$

within a search space defined by $\Delta\sigma_\uparrow, \Delta\sigma_\downarrow \in [0, 0.5]$. The above listed constraints exist to ensure that $\dot{\sigma}(t)$ is bounded from below by zero and from above by the extended line defining the central segment of the trajectory profile (*i.e.* the dash-dotted line in Figure 6.3). This is not the case, for instance, in Figure 6.3, where the jerk's (*i.e.* $\dddot{\sigma}(t)$) sign has been labeled at various points of interest along the trajectory profile. Through observation, one may conclude that the undesirable situations that are illustrated require the jerk to undergo two sign changes within the ramp-up and/or ramp-down segments. It is known from Eqs. (6.3) and (6.4) that $\dddot{\sigma}(t)$ is a quadratic polynomial which would allow for these sign changes to occur. However, by imposing $\text{sign}[\dddot{\sigma}(0)] = -\text{sign}[\dddot{\sigma}(t_\uparrow)]$ and $\text{sign}[\dddot{\sigma}(t_f - t_\downarrow)] = -\text{sign}[\dddot{\sigma}(t_f)]$ through the constraints in Eq. (6.16), this is made impossible. In addition to this, it is known that for $\dot{\sigma}(t)$ to be non-negative with $\ddot{\sigma}(0) = \ddot{\sigma}(t_f) = 0$ then $\dddot{\sigma}(0) \geq 0$ and $\dddot{\sigma}(t_f) \geq 0$ are necessary conditions as are imposed by constraints g_1 and g_4 in Eq. (6.16). In order to obtain the preliminary trajectory profile, the sequential-quadratic programming (SQP) algorithm implemented in MATLAB's `fmincon` function may be used to solve the optimization problem (although other algorithms could be substituted for greater efficiency).

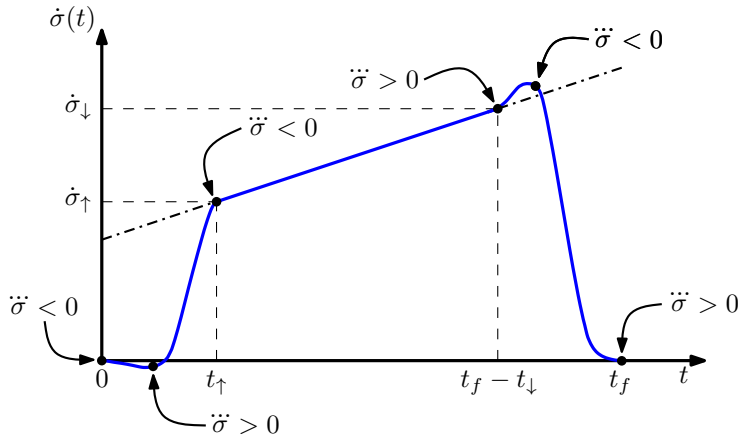


Figure 6.3: Illustration of unwanted characteristics within the trajectory profile (labels related to $\dddot{\sigma}$ pertain only to the ramp-up/-down sections of the trajectory profile since $\dddot{\sigma} = 0$ throughout the central segment).

6.5 Trajectory optimization: Phase II

In Phase I of the proposed trajectory optimization methodology, a preliminary trajectory was obtained by minimizing the trajectory duration while ensuring that the rockbreaker's flow rate capacity was fully exploited at the end of the ramp-up segment and the beginning of the ramp-down segment. In Phase II, this trajectory will be modified so that the flow rate capacity is fully exploited in the central trajectory segment. To facilitate this process, the trajectory will henceforth be represented in discrete form over a series of N time instants $r\Delta t$ ($r = 0, 1, \ldots, N-1$) separated by a constant increment Δt so that the $0 \leq t \leq t_f$ time interval is completely covered (*i.e.* $t_f = (N-1)\Delta t$). This has the added benefit of naturally leading to the generation of the desired trajectory in discrete form for the purpose of implementation within a control algorithm.

6.5.1 Computation of the normalized flow rate vector

The preliminary trajectory is obtained from Phase I in terms of parameters $t_{\uparrow}$, $t_{\downarrow}$, $\Delta\sigma_{\uparrow}$, $\Delta\sigma_{\downarrow}$, $\dot{\sigma}_{\uparrow}$ and $\dot{\sigma}_{\downarrow}$. Through the boundary conditions expressed in Eqs. (6.5) and (6.6), these parameters lead to the piecewise definition of $\sigma(t)$ provided in Eqs. (6.3), (6.4) and (6.7) which, along with its time derivative, may be converted to discrete form as $\sigma(r\Delta t)$ and $\dot{\sigma}(r\Delta t)$. Referring to Eqs. (6.11), (3.15) and (3.19), one has $\dot{\mathbf{x}}(r\Delta t) = \mathbf{e}\cdot\dot{\sigma}(r\Delta t)$, $\dot{\boldsymbol{\rho}}(r\Delta t) = \mathbf{J}\dot{\mathbf{x}}(r\Delta t)$ and $\mathbf{Q} = \mathbf{J}_q|\dot{\boldsymbol{\rho}}(r\Delta t)|$, respectively, where $\mathbf{J}$ and $\mathbf{e}$ are also dependent on $\sigma(r\Delta t)$. The vector of normalized flow rates is then obtained (similar

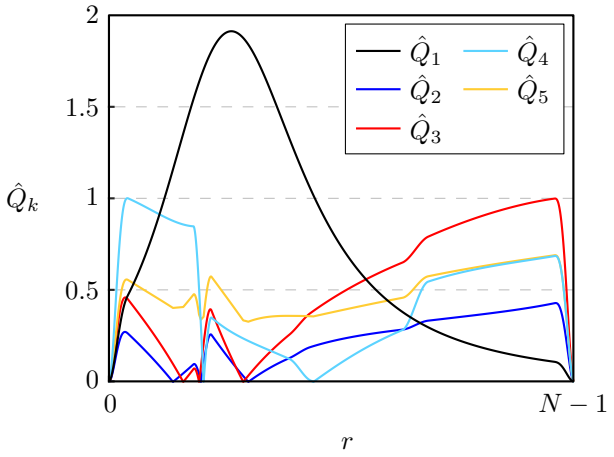


Figure 6.4: Normalized flow rates as a function of the time increments r for a typical rockbreaker trajectory.

to Eq. (6.13)) as

$$\hat{\mathbf{Q}}(r\Delta t) = \left[\frac{Q_1(r\Delta t)}{Q_{1_{\max}}}\ ,\ \frac{Q_2(r\Delta t)}{Q_{2_{\max}}}\ ,\ \frac{Q_3(r\Delta t)}{Q_{3_{\max}}}\ ,\ \frac{Q_4(r\Delta t)}{Q_{4_{\max}}}\ ,\ \frac{Q_p(r\Delta t)}{Q_{p_{\max}}}\right]^T \tag{6.17}$$

The variations in time of the normalized flow rates (in terms of the time increments r) for a sample rockbreaker trajectory are shown in Figure 6.4 where it should be noted that: i) Situations where $\hat{Q}_k > 1$ indicate a violation of the flow rate limits and ii) abrupt changes that occur when $\hat{Q}_k = 0$ are due to the fact that the absolute values of the flow rates are being used and thus do not correspond to a lack of smoothness in flow rate variations.

6.5.2 Determination of the scaling ratio function

As has been previously mentioned, the minimization of the trajectory duration goes hand-in-hand with the full use of the rockbreaker's flow rate capacity. This is to be accomplished by scaling the rockbreaker's flow rates up or down based on a scaling ratio $\lambda(r\Delta t)$ that varies with time. The determination of this scaling ratio as a function of time is the purpose of this section while the details regarding its use are discussed in the next section.

Given the rockbreaker is at rest at the beginning and end of each planned trajectory, the full exploitation of its flow rate capacity is not possible nor desirable during the ramp-up/-down segments. With this in mind, Phase I of the trajectory optimization has already produced a preliminary trajectory where at least one of the four proportional valves or the hydraulic pump is at its flow rate

limit at both $t_\uparrow$ and $t_f - t_\downarrow$, *i.e.* $\max(\hat{Q}_{k_\uparrow}) = \max(\hat{Q}_{k_\downarrow}) = 1$. However, this does not guarantee that flow rate limits are respected throughout the ramp-up/-down segments and this must be considered in the determination of the scaling ratio function. With the exception of the ramp-up/-down segments of the trajectory, the scaling ratio function is determined based on fully utilizing the flow rate capacities. The detailed procedure that is used to create the $\lambda(r\Delta t)$ function is as follows:

1. **Identify the maximum normalized flow rates within the ramp-up/-down segments:** The time increments corresponding to the end of the ramp-up segment and the beginning of the ramp-down segment are first defined as $r_\uparrow \Delta t$ and $r_\downarrow \Delta t$, respectively, with $r_\uparrow = \max(r) \mid r\Delta t \leq t_\uparrow$ and $r_\downarrow = \min(r) \mid r\Delta t \geq t_f - t_\downarrow$. The maximum normalized flow rate within the ramp-up segment may then be identified as

$$\hat{Q}_a = \max_k \left\{ \max_{0 \leq r \leq r_\uparrow} \left[\hat{Q}_k(r\Delta t) \right] \right\} \tag{6.18}$$

   where $\hat{Q}_k(r\Delta t)$ is the k^{th} element of $\hat{\mathbf{Q}}(r\Delta t)$ and where the maximum is denoted to occur at the ${r_a}^{\text{th}}$ time increment. Similarly, the maximum normalized flow rate within the ramp-down segment is

$$\hat{Q}_b = \max_k \left\{ \max_{r_\downarrow \leq r \leq N-1} \left[\hat{Q}_k(r\Delta t) \right] \right\} \tag{6.19}$$

   which occurs at the ${r_b}^{\text{th}}$ time increment. It is noted that, more often than not, one will find $r_a = r_\uparrow$, $r_b = r_\downarrow$, $\hat{Q}_a \approx 1$ and $\hat{Q}_b \approx 1$ given how the preliminary trajectory profile was generated. The trajectory whose normalized flow rates are plotted in Figure 6.4 is an example of a situation where this is not the case. In order to emphasize this fact, a close-up view of a region of the plot in Figure 6.4 has been provided in Figure 6.5 where it may be seen that $\hat{Q}_a > 1$.

2. **Define the scaling ratio function over the initial and final trajectory segments:** While the rockbreaker's velocity is ramped-up/-down over the time intervals corresponding to $0 \leq r \leq r_\uparrow$ and $r_\downarrow \leq r \leq N-1$, respectively, it may be said based on observations made in the description of the previous step that its flow rates ramp-up/-down over the time intervals corresponding to $0 \leq r \leq r_a$ and $r_b \leq r \leq N-1$. During the latter, the rockbreaker's flow rate capacity cannot be fully exploited as this would entail sudden increases (decreases) in the flow

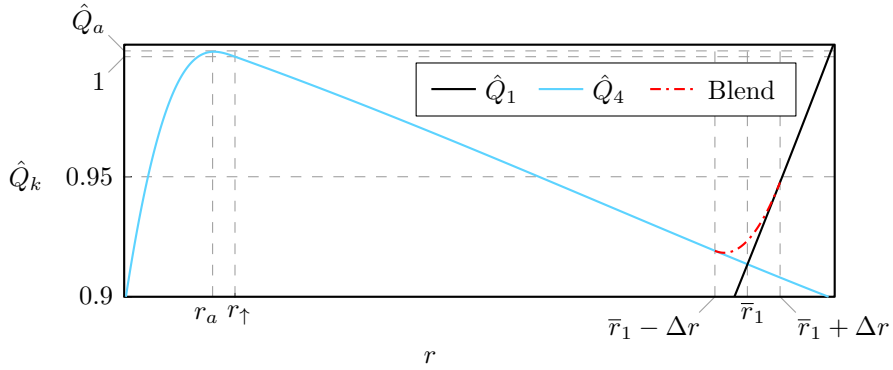


Figure 6.5: Close-up view of a region of Figure 6.4.

rates at the beginning (end) of the trajectory. With this in mind, and to ensure that flow rate limits are respected, the scaling ratio function over these time intervals is defined to remain constant as follows:

$$\lambda(r\Delta t) = \frac{1}{\hat{Q}_a} \quad \text{when} \quad 0 \leq r \leq r_a \tag{6.20}$$

$$\lambda(r\Delta t) = \frac{1}{\hat{Q}_b} \quad \text{when} \quad r_b \leq r \leq N-1 \tag{6.21}$$

Moreover, since it is known that $\hat{Q}_a \geq 1$ and $\hat{Q}_b \geq 1$, this will not shorten the duration of the rockbreaker's ramp-up/-down trajectory segments. Since the initial durations (*i.e.* $t_\uparrow$ and $t_\downarrow$) were empirically chosen to be long enough to maintain the rockbreaker's accelerations at reasonable levels, this is a desirable feature of the proposed approach.

3. **Define the scaling ratio function for the remainder of the trajectory:** Following the previous step, it is known that at least one of the proportional valves or the hydraulic pump will be operating at its flow rate limit at the time increments corresponding to r_a and r_b once the trajectory scaling is complete. For the trajectory segment defined by $r_a < r < r_b$, it is then possible to scale the flow rates so that the rockbreaker's flow rate capacity is fully exploited. This leads to the following definition of the scaling ratio function over the remaining trajectory segment:

$$\lambda(r\Delta t) = \frac{1}{\max\limits_{k} \left[\hat{Q}_k(r\Delta t)\right]} \quad \text{with} \quad r_a < r < r_b \tag{6.22}$$

Over the course of the trajectory, the hydraulic system component (*i.e.* proportional valve or pump) responsible for constraining the rockbreaker's motion will vary. For example, in

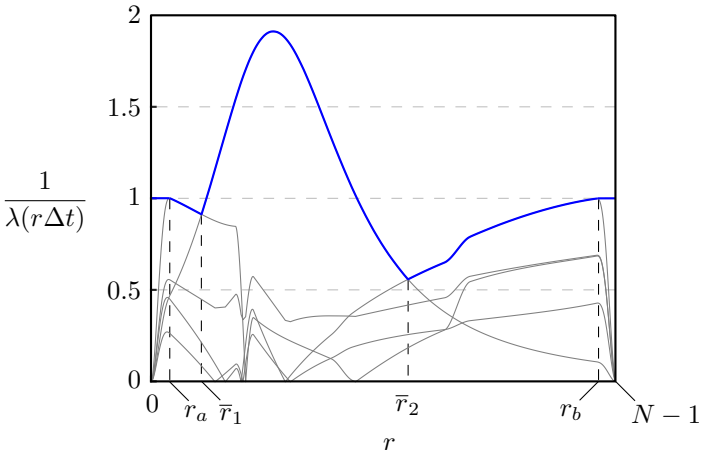


Figure 6.6: Inverse of scaling ratio function for complete rockbreaker trajectory. Note that the trajectory is the same as was used to produce Figure 6.4.

Figure 6.5, the proportional valve associated to ρ_4 is initially the most constraining from r_a to $\overline{r}_1$ at which point the valve associated to ρ_1 momentarily takes over the role. The time increments at which these transitions occur are denoted as $\overline{r}_1$, $\overline{r}_2$, etc. Based on Eqs. (6.20) through (6.22), the scaling ratio function is fully defined and its inverse may thus be plotted as a function of r as is done in Figure 6.6. In this plot, the original normalized flow rates are also included (in gray color) to emphasize the meaning of the $\lambda(r\Delta t)$ function.

4. **Add parabolic blends to smooth the scaling ratio function:** At this point in the process, the scaling ratio function $\lambda(r\Delta t)$ has been defined for the entire trajectory duration. While $\lambda(r\Delta t)$ (or its inverse which is plotted in Figure 6.6) is necessarily continuous, it will typically not be smooth given the transitions occurring at r_a, r_b, $\overline{r}_1$, $\overline{r}_2$, etc.[2] It will become clear in Section 6.5.3 that this constitutes an issue since the flow rate scaling, which depends on $\lambda(r\Delta t)$, has a direct impact on the smoothness of $\dot{\sigma}(t)$. Moreover, if $\dot{\sigma}(t)$ is not a smooth function then the rockbreaker accelerations will not be continuous. For this reason, polynomial blends are added to the inverse of the scaling ratio function. The transition occurring at $\overline{r}_1$ in Figure 6.5 will be used to illustrated the implementation of the blends where in that particular case $\hat{Q}_1$ and $\hat{Q}_4$ are to be blended together. The blend duration is first set empirically to t_{blend} (*e.g.* $t_{\text{blend}} = 0.2$ seconds). The quantity of time increments over which the blend is applied

[2] In this context, the terms "continuous" and "smooth" are applied loosely when referring to the scaling ratio function since it is technically nonsensical to discuss such qualities as they relate to a discrete function.

on either side of $\overline{r}_1$ may then be computed as $\Delta r = t_{\text{blend}}/(2\Delta t)$ where the result is rounded down to the nearest integer. This duration is shortened to exclude any peaks in the $\hat{Q}_k$ being blended (there are none in the particular case currently being illustrated). Once a suitable blend time span has been identified, a 3rd-order polynomial blend $\eta(r\Delta t)$, centered on $\overline{r}_1$, is determined based on the following boundary conditions:

$$\eta\big[(\overline{r} - \Delta r)\Delta t\big] = \hat{Q}_{k_1}\big[(\overline{r} - \Delta r)\Delta t\big] \tag{6.23}$$

$$\eta\big[(\overline{r} + \Delta r)\Delta t\big] = \hat{Q}_{k_2}\big[(\overline{r} + \Delta r)\Delta t\big] \tag{6.24}$$

$$\dot{\eta}\big[(\overline{r} - \Delta r)\Delta t\big] = \frac{\hat{Q}_{k_1}\big[(\overline{r} - \Delta r + 1)\Delta t\big] - \hat{Q}_{k_1}\big[(\overline{r} - \Delta r - 1)\Delta t\big]}{2\Delta t} \tag{6.25}$$

$$\dot{\eta}\big[(\overline{r} + \Delta r)\Delta t\big] = \frac{\hat{Q}_{k_2}\big[(\overline{r} - \Delta r + 1)\Delta t\big] - \hat{Q}_{k_2}\big[(\overline{r} - \Delta r - 1)\Delta t\big]}{2\Delta t} \tag{6.26}$$

where k_1 and k_2 represent the indices associated to each of the normalized flow rates being blended while $\overline{r}$ is the initial location of the transition between the latter (in the illustration $\overline{r} = \overline{r}_1$, $k_1 = 4$ and $k_2 = 1$). In Eqs. (6.25) and (6.26), the centered difference is used to estimate the slope of the $\hat{Q}_k$ functions at the blend's boundaries. The result is a smooth blend such as the one shown in Figure 6.5. Implementing this approach at all transition points, a smooth scaling ratio function is finally obtained.

6.5.3 Trajectory scaling

The scaling ratio function (*i.e.* $\lambda(r\Delta t)$), which has now been fully developed, is to be used to scale the rockbreaker's flow rates (*i.e.* $\mathbf{Q}$) in order to minimize trajectory durations. However, based on the developments in Section 6.5.1, it is known that $\mathbf{Q}$ and $\dot{\sigma}$ are related linearly. It is thus equivalent and more convenient to apply the scaling ratio function directly to $\dot{\sigma}$. For this purpose, it is recalled that $\dot{\sigma}(r\Delta t)$ is a discrete function which essentially consists of a list of values $\dot{\sigma}_r$ to which are associated corresponding times $t_r = r\Delta t$ with $r = 0, 1, \ldots, N-1$. At the r^{th} time increment, σ_r may be scaled as follows:

$$\dot{\tilde{\sigma}}_r = \lambda(r\Delta t) \cdot \dot{\sigma}_r \tag{6.27}$$

which, as previously mentioned, has a direct impact on the rockbreaker's velocity along the trajectory. In order for the rockbreaker's displacements during the trajectory to remain unchanged despite the changes in velocity, the scaling must also be applied to the time series t_r. If $\Delta t_r = \Delta t$ is defined

as the initial time span separating the ${t_{r-1}}^{\text{th}}$ and ${t_r}^{\text{th}}$ time increments such that $t_r = t_{r-1} + \Delta t_r$ then the scaled version becomes

$$\check{t}_r = \check{t}_{r-1} + \Delta\check{t}_r \tag{6.28}$$

with

$$\Delta\check{t}_r = \frac{2\Delta t_r}{\lambda(r\Delta t) + \lambda[(r-1)\Delta t]} \tag{6.29}$$

where in this case $r = 1, 2, \ldots, N-1$. The displacement level trajectory is then simply obtained by associating to the original list of discrete values σ_r (*i.e.* obtained at the discrete time increments t_r using Eqs. (6.3), (6.4) and (6.7)) new time increments $\check{t}_r$. This means that $\check{\sigma}_r = \sigma_r$ but the values occur at different instants in time (*i.e.* $\check{t}_r$ rather than t_r).

6.6 Post-processing of prescribed actuator motion data

Referring to Eqs. (6.28) and (6.29), it may be observed that the time histories of $\check{\sigma}$ and $\dot{\check{\sigma}}$ that result from the trajectory scaling process are based on a time series with variable time increments. This undesirable situation may be rectified by linearly interpolating between consecutive values of $\check{\sigma}$ and $\dot{\check{\sigma}}$, respectively, combined with the corresponding time data. The result is a series of $\check{\sigma}$ and $\dot{\check{\sigma}}$ values associated to a time series with a uniform time increment that is typically based on the frequency of the semi-automatic rockbreaker system's control loop. When using such an approach, it is advisable to generate the initial trajectory based on time increments an order of magnitude smaller than those that will eventually be required for the purpose of the control algorithm implementation as this will improve the accuracy of the linear interpolation. The hydraulic actuator lengths and rates of change associated to $\check{\sigma}$ and $\dot{\check{\sigma}}$, respectively, may then be computed based on the kinematic relationships developed in Sections 3.3, 3.4 and 3.6.

6.7 Simulation results

In order to compare the effectiveness of the aforementioned trajectory planning algorithm (labeled as Method 2) to generate time-optimal trajectories, a Cartesian trajectory planning algorithm based on the methodology presented in Chapter 5 (labeled as Method 1) was developed. The trajectories generated using both methods were based on the rockbreaker system previously used in Chapter 5,

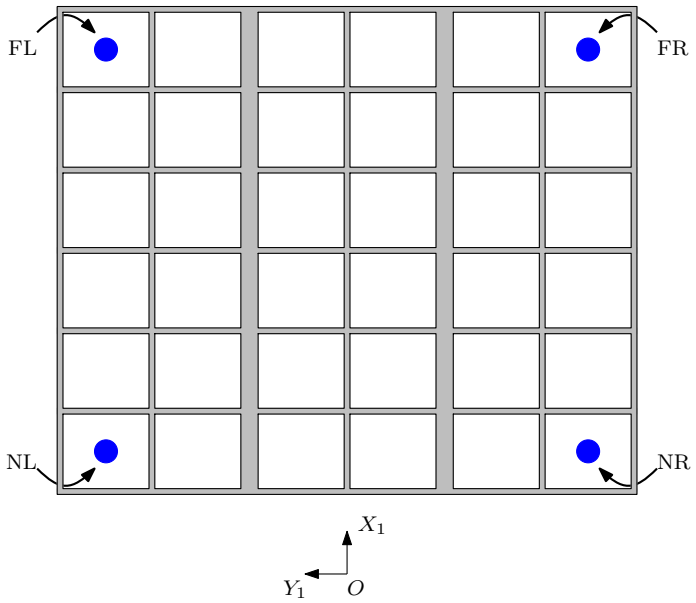


Figure 6.7: Top down view of the grizzly with the initial and final rockbreaker configurations used for trajectory planning algorithm validation. Note that the grizzly installation shown is not to scale.

whose parameters can be obtained from Appendix B and where $\varphi = 5.20°$ and $Q_{p_{\max}} = 170$ L/min. Furthermore, the value of $t_{\uparrow}$ and $t_{\downarrow}$ is set to 2 seconds (to satisfy the conditions given in Section 6.4.3), while t_{blend} is set to 1 second. The value for t_{blend} is set empirically and may be changed as required based on test results.

Six case trajectories, whose initial and final rockbreaker configurations are defined based on a set of four points located in the work plane above the grizzly, were generated using both trajectory planning methods. The points about which the rockbreaker's tip is moved represent the centres of the four corner openings of the grizzly, as shown in Figure 6.7, where the grizzly is symmetrically positioned with respect to the rockbreaker's X_1Z_1 plane. These points are listed as near left (NL), near right (NR), far left (FL) and far right (FR). The configuration of the rockbreaker for each point are given as:

$$\mathbf{x}_{\text{NL}} = \begin{bmatrix} 2.20\,\text{m} \\ 2.15\,\text{m} \\ -1.00\,\text{m} \\ 270° \end{bmatrix}, \quad \mathbf{x}_{\text{NR}} = \begin{bmatrix} 2.20\,\text{m} \\ -2.15\,\text{m} \\ -1.00\,\text{m} \\ 270° \end{bmatrix}, \quad \mathbf{x}_{\text{FL}} = \begin{bmatrix} 6.27\,\text{m} \\ 2.15\,\text{m} \\ -1.00\,\text{m} \\ 293.6° \end{bmatrix}, \quad \mathbf{x}_{\text{FR}} = \begin{bmatrix} 6.27\,\text{m} \\ -2.15\,\text{m} \\ -1.00\,\text{m} \\ 293.6° \end{bmatrix}$$

The generated trajectories were defined as:

Case 1. Moving from the initial position NL to the final position FR.

Case 2. Moving from the initial position FR to the final position NL.

Case 3. Moving from the initial position NL to the final position FL.

Case 4. Moving from the initial position FL to the final position NL.

Case 5. Moving from the initial position NL to the final position NR.

Case 6. Moving from the initial position FL to the final position FR.

With these six trajectories, all possible trajectory behaviours between a set of two corner grid openings of a grizzly are obtained. This is part due to the symmetry of the point locations and swing joint position. If one were to compute all complimentary trajectories (*e.g.* trajectory 1 goes from NR to FL rather than NL to FR), one would observe identical actuation for the actuators responsible for the motion of the planar arm of the rockbreaker, whilst the swing joint's actuators would exhibit their complimentary behaviours, *i.e.* the displacement for ρ_0 would be that of ρ_1 and vice versa.

The results for Case 1 in terms of $\dot{\sigma}(t)$ are shown in Figure 6.8. It may be observed that Method 1 leads to a constant $\dot{\sigma}$ for the majority of its trajectory (*i.e.* a constant Cartesian straight-line velocity) and has a total duration of 32.84 second. Meanwhile, Method 2 exhibits a fluctuating $\dot{\sigma}$ throughout the trajectory and has a total duration of 18.93 seconds, which is significantly shorter. A comparison of the resulting ρ_j trajectories of Methods 1 and 2 is illustrated in Figures 6.9-6.13. Furthermore, plots of all ρ_j trajectories for each of Methods 1 and 2 are provided in Figures 6.14 and 6.15.

When comparing the utilization of flow rates (using the normalized flow rates) amongst both Methods 1 and 2, as illustrated in Figures 6.16 and 6.17, respectively, it is observed that both methods respect the system's flow rate limitations. However, from Figure 6.16 it is evident that the flow rate usage

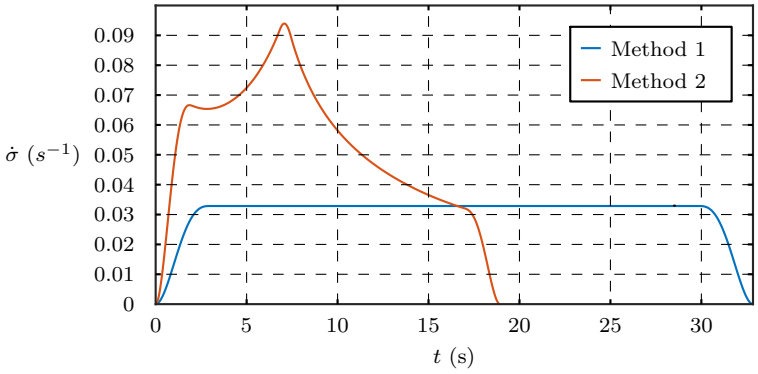


Figure 6.8: Comparison of the resulting $\dot{\sigma}$ trajectory from both trajectory planning methods for Case 1.

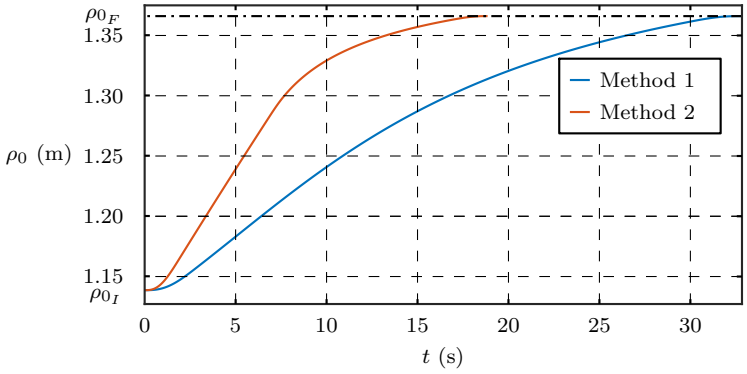


Figure 6.9: Comparison of the resulting ρ_0 trajectory from both trajectory planning methods for Case 1.

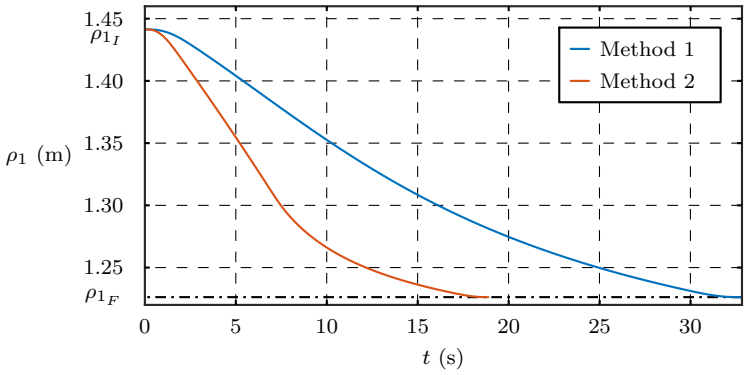


Figure 6.10: Comparison of the resulting ρ_1 trajectory from both trajectory planning methods for Case 1.

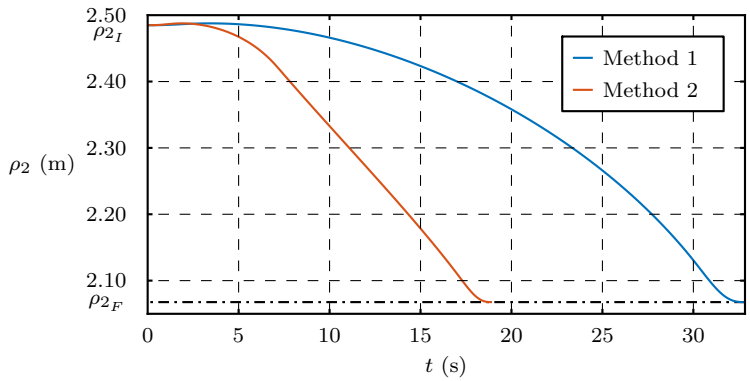


Figure 6.11: Comparison of the resulting ρ_2 trajectory from both trajectory planning methods for Case 1.

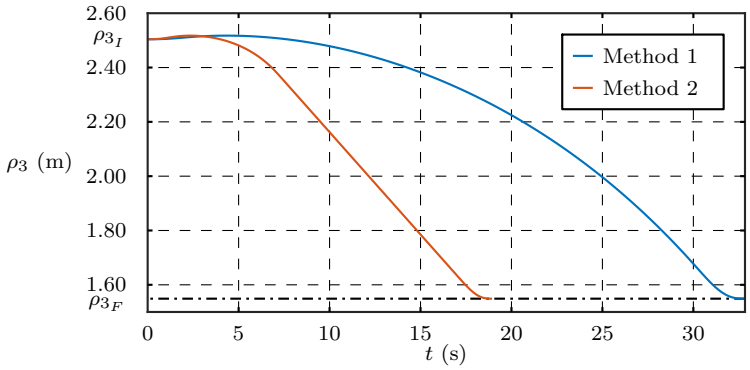


Figure 6.12: Comparison of the resulting ρ_3 trajectory from both trajectory planning methods for Case 1.

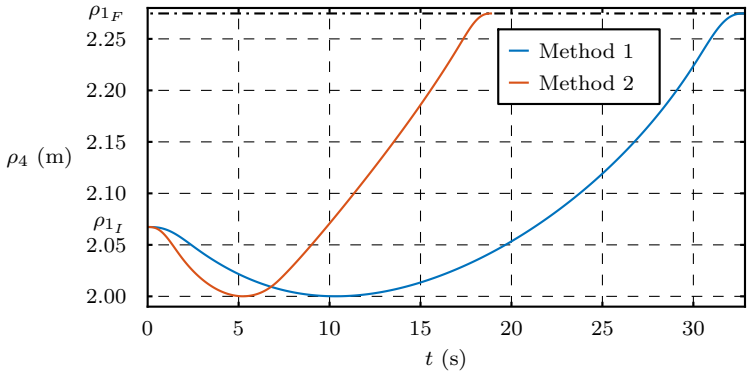


Figure 6.13: Comparison of the resulting ρ_4 trajectory from both trajectory planning methods for Case 1.

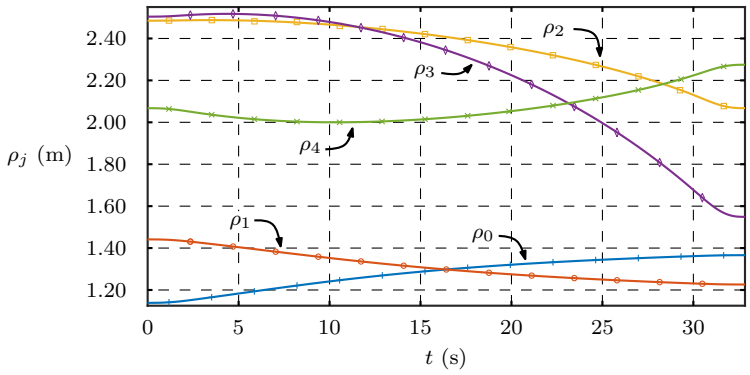


Figure 6.14: Resulting ρ_j trajectories obtained using trajectory planning Method 1 for Case 1.

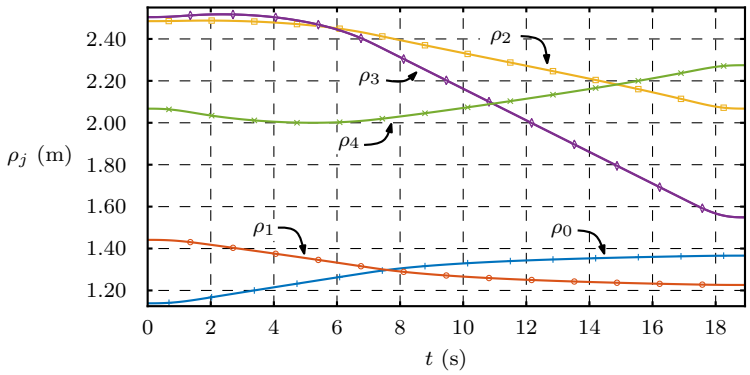


Figure 6.15: Resulting ρ_j trajectories obtained using trajectory planning Method 2 for Case 1.

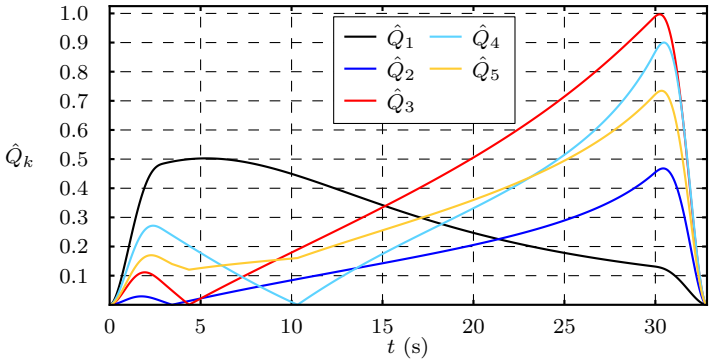


Figure 6.16: Resulting $\hat{Q}_k$ trajectory from sub-optimal trajectory planning method (Method 1) for Case 1.

throughout the trajectory is not optimized by Method 1, as it only reaches the maximum allowable flow rate $Q_{3_{\max}}$ at a given time instant. On the other hand, it may be seen in Figure 6.17 that Method 2 fully exploits available flow rates throughout the majority of its generated trajectory, resulting in a shorter duration of the latter.

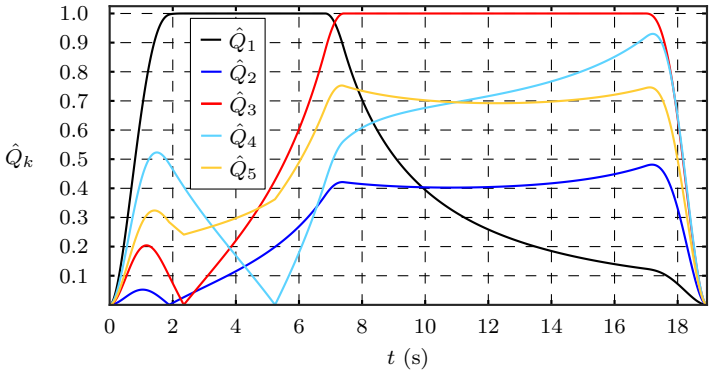


Figure 6.17: Resulting $\hat{Q}_k$ trajectory from flow optimal trajectory planning method (Method 2) for Case 1.

Table 6.1: Total duration of trajectories using two trajectory planning methods.

Case	t_f (s) - Method 1	t_f (s) - Method 2	Reduction in t_f (%)
1	32.84	18.93	42.4
2	50.03	24.75	50.5
3	22.94	14.48	36.9
4	34.46	20.98	39.1
5	19.28	15.39	20.2
6	15.09	10.17	32.6
Average	29.11	17.45	37.0

Although detailed results have only been provided for Case 1, it should be noted that the remaining cases exhibit similar outcomes when comparing both trajectory planning methods. However, the resulting motion duration of the trajectories for all cases using both methods is provided in Table 6.1, where it can be seen that Method 2 generates trajectories which are of shorter duration, making these time-optimal in comparison to trajectories generated using Method 1.

6.8 Discussion and conclusions

In summary, the newly proposed trajectory planning algorithm presented within this chapter allows for the generation of time-optimized trajectories which are smooth and continuous. This is accomplished through a scaling of the velocity-level trajectory so as to ensure a maximum utilization of the available flow rates by the actuators for the central trajectory segment, which excludes the initial and final velocity ramping segments.

The resulting trajectories using this new approach are, on average, 37% faster than the trajectories generated using the method presented in Chapter 5. This is a direct result of the full utilization of allowable flow rates throughout the bulk of the trajectory as compared to the previous approach which only ensured full flow rate utilization for a single time instant. However, it should be noted that the reductions in trajectory durations are more pronounced when the motion of the arm within the X_2Y_2 plane (refer to Figure 3.2(b)) is substantial. Furthermore, one could use this approach to generate a trajectory akin to a trajectory with via points by generating trajectory between these via points and compounding them as one trajectory. However, this approach would not be time-optimal, seeing as the rockbreaker's motion would stop momentarily during the transitions between trajectory segments. Furthermore, the main drawback from using this trajectory planning algorithm (as well as the technique presented in Chapter 5) is its offline computation of trajectories, preventing any alterations to trajectories during the rockbreaker's motion to be made, which could be problematic if this trajectory planner was applied in a real-time collision avoidance scheme.

Further improvements regarding this algorithm would require the availability of an operational system for testing purposes. With such a system, one could find an optimal ramping motion duration value ($t_\uparrow$ and $t_\downarrow$) to be used in the trajectory planning algorithm, as these were set empirically during the simulations (see Section 6.4.3). Furthermore, although in theory the system should be able to supply the required flow rates for the motions generated, this assumption should be validated. Moreover, since the generated trajectories ensure that one of the actuators is saturated during any motion (*i.e.* $\hat{Q}_k$ computed based on $Q_{k_{\max}}$), a controller may not be able to compensate for small positional errors as additional flow rate to one of the actuators may not be feasible. If testing reveals this to be the case, a reduction of the flow rate limits may be imposed on the trajectory planning algorithm (*i.e.* $\hat{Q}_k$ computed based on $Q_{k_{\max}} = 0.9Q_{k_{\max}}$). Furthermore, from Figure 5.3, it can be shown that as θ_1 nears the extremities of its range ($\pm 85°$), the angular velocity $\dot{\theta}_1$ increases exponentially while flow rate remains constant. Even with a reduction of the flow rate limit for this joint, motions where θ_1 nears those positions would lead to a dramatic increase in joint velocity, making it difficult to control. Should testing reveal this to be problematic, an alternative solution is proposed, where $\hat{Q}_1$ is computed based on a maximum allowable swing joint velocity $\dot{\theta}_{1_{\max}}$ rather than its maximum allowable flow rate $Q_{1_{\max}}$. This would effectively generate trajectories where

flow rate is exponentially reduced as θ_1 nears its extremities, reducing the likelihood of positional overshoot during any swing joint motion. Lastly, future research regarding trajectory planning algorithms for a rockbreaker would entail the development of online motion planners that ensure smooth and continuous motions, such that they may be used in an online collision-avoidance scheme.

Chapter 7

Collision-free trajectory planning

In Chapters 5 and 6, robust trajectory planning algorithms that may be used for a semi-autonomous rockbreaker system were described. However, as was mentioned in Section 2.3, these algorithms do not account for interactions between the rockbreaker and its environment and, therefore, will generate trajectories that may not be collision-free. The work presented within this chapter seeks to mitigate this issue for automated rockbreaker motions within its expected work environment (*i.e.* within the work plane) by utilizing these trajectory planning algorithms within a collision avoidance strategy. Initially, a generated trajectory is evaluated for possible collisions between the rockbreaker's links as well as obstacles found within its environment by using the enhanced Gilbert-Johnson-Keerthi (GJK) algorithm. If no collisions are found, the trajectory may safely be sent to the controller to move the rockbreaker accordingly. However, if collisions are detected, the collision avoidance strategy is run to generate a new, collision-free trajectory.

The algorithms and logic used for this collision avoidance strategy will be described within the remainder of this chapter, which is organized as follows: an overview of the enhanced GJK algorithm is provided in Section 7.1, followed by the modeling techniques and collision pair identification description in Section 7.2. The collision avoidance strategies used in conjunction with the trajectory planning algorithms are then discussed in Section 7.3, followed by the results of the collision avoidance strategy's implementation in a simulation provided in Section 7.4. Lastly, a general discussion and conclusion regarding this strategy is given in Section 7.5.

7.1 Overview of the enhanced Gilbert-Johnson-Keerthi algorithm

Although the GJK algorithm and its collision detecting variation have been defined extensively in the literature, a brief overview of the algorithm is provided here. The GJK algorithm is a fast and robust method to compute the Euclidean distance between a pair of convex objects within a real coordinate space of dimension m, *i.e.* $\mathbb{R}^m$ [58]. That being said, it is more efficient for cases where $m = 3$ and where the pair of convex objects are described using polytopes with defined vertices [58]. If the Euclidean distance between a pair of objects is found to be null, an intersection between the objects has occurred. Based on this principle, the GJK algorithm may be used as a collision detection algorithm. Furthermore, since collision detection only relies on Boolean logic (where the Euclidean distance either is or isn't equal to zero), the GJK algorithm may be simplified, yielding what is referred to as the enhanced GJK algorithm.

Let $\mathcal{A}$ and $\mathcal{B}$ define the sets of points which make up two convex polytopes, as shown in Figure 7.1(a). The Euclidean distance between these polytopes, $d(\mathcal{A}, \mathcal{B})$, may be computed as:

$$d(\mathcal{A}, \mathcal{B}) = \min\{\|\mathbf{a} - \mathbf{b}\| : \mathbf{a} \in \mathcal{A}, \mathbf{b} \in \mathcal{B}\} \tag{7.1}$$

where $\mathbf{a}$ and $\mathbf{b}$ represent the positions of points located within $\mathcal{A}$ and $\mathcal{B}$, respectively, expressed in a XYZ reference frame with origin O. The Minkowski sum of $\mathcal{A}$ and $-\mathcal{B}$, *i.e.*

$$\mathcal{C} = \mathcal{A} \oplus (-\mathcal{B}) = \{\mathbf{a} - \mathbf{b} \ : \ \mathbf{a} \in \mathcal{A}, \mathbf{b} \in \mathcal{B}\} \tag{7.2}$$

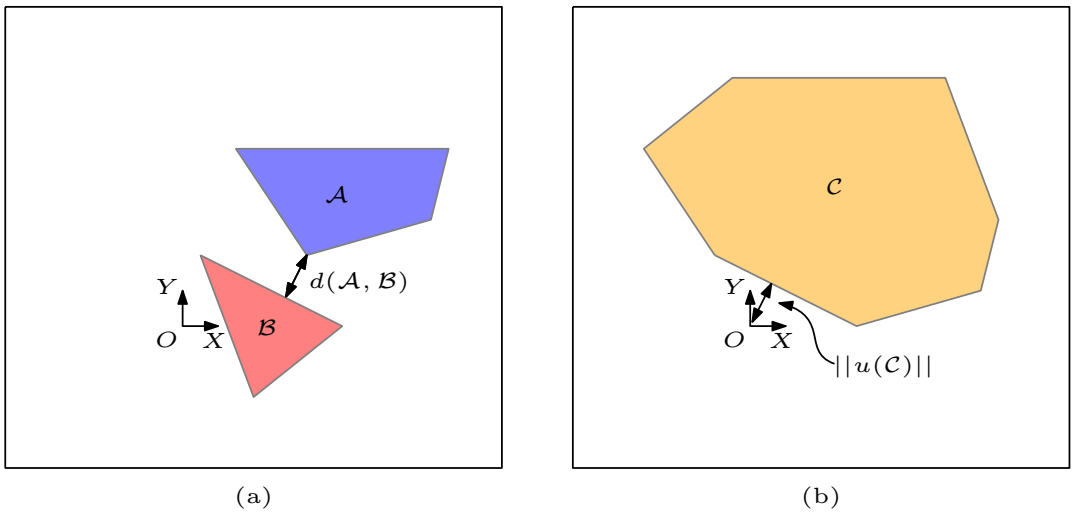
is the convex polytope, shown in Figure 7.1(b), that encompasses the set of points resulting from all possible difference operations between points in $\mathcal{A}$ and $\mathcal{B}$. The Euclidean distance between the polytopes defined by $\mathcal{A}$ and $\mathcal{B}$ may then be obtained as:

$$d(\mathcal{A}, \mathcal{B}) = \|u(\mathcal{C})\| \tag{7.3}$$

where $\|u(\mathcal{C})\|$ is the minimal distance from O of all points within $\mathcal{C}$, *i.e.*

$$\|u(\mathcal{C})\| = \min\{\|\mathbf{c}\|, \mathbf{c} \in \mathcal{C}\} \tag{7.4}$$

Note that the function $\|u(\cdot)\|$ is defined here for any set of points found within the convex object specified as the function's argument. Since $\|u(\mathcal{C})\| = 0$ corresponds to a collision between the polytopes defined by $\mathcal{A}$ and $\mathcal{B}$, this infers that the set of points $\mathcal{C}$ includes the origin O.



(a) (b)

Figure 7.1: Euclidean distance computation between two convex objects in $\mathbb{R}^2$: (a) Polytopes $\mathcal{A}$ and $\mathcal{B}$ separated by a distance $d(\mathcal{A},\mathcal{B})$ and (b) the resulting set of the Minkowski sum, $\mathcal{C}$, where $\|u(\mathcal{C})\| = d(\mathcal{A},\mathcal{B})$.

In order to obtain $u(\mathcal{C})$, the Minkowski sum defined in Eq.(7.2) must be found. However, computing the entire Minkowski sum between objects made up of numerous points can be computationally expensive. From Carathéodory's theorem [68], it is known that if a point lies within a convex object in $\mathbb{R}^m$, the number of points required to generate a convex hull which surrounds this point is no greater than $m+1$. This is the basis of the enhanced GJK algorithm. It iteratively seeks to identify if a simplex, made up of no more than $m+1$ points found within $\mathcal{C}$, includes the origin O. In the k^{th} iteration of the enhanced GJK algorithm, a point W_k of position $\mathbf{w}_k$ is added to the set of points $\mathcal{W}_k$ in an attempt to generate a simplex which includes O. The support function $s_{\mathcal{C}}$ yields the point W_k, located on the perimeter of $\mathcal{C}$, which is the furthest from the origin O along a direction specified by $-\mathbf{v}_k$, *i.e.*

$$s_{\mathcal{C}}(-\mathbf{v}_k) = \mathbf{w}_k \;\Big|\; -\mathbf{v}_k \cdot \mathbf{w}_k = \max_{\mathbf{c}}\{-\mathbf{v}_k \cdot \mathbf{c} \;:\; \mathbf{c} \in \mathcal{C}\} \tag{7.5}$$

In essence, the output for the support function $s_{\mathcal{C}}(-\mathbf{v}_k)$ is equal to the vector $\mathbf{c}$ (found within $\mathcal{C}$) which provides the maximum numerical value when taking the dot product between $\mathbf{c}$ and $-\mathbf{v}_k$. Note that, since the polytope defined by the sets of points $\mathcal{C}$ is convex, Eq. (7.5) will always return a point which lies along the perimeter of this polytope. Since $\mathcal{C}$ is the resulting set of points derived from the Minkowski sum of $\mathcal{A}$ and $-\mathcal{B}$, the support function $s_{\mathcal{C}}(-\mathbf{v}_k)$ can be described from the

support functions and mapping of the polytopes (defined by $\mathcal{A}$ and $\mathcal{B}$) such that:

$$s_{\mathcal{C}}(-\mathbf{v}_k) = s_{\mathcal{A}}(-\mathbf{v}_k) - s_{\mathcal{B}}(\mathbf{v}_k) \tag{7.6}$$

Using this form of support function eliminates the need to compute set $\mathcal{C}$ in its entirety.

With support functions having now been defined, the enhanced GJK algorithm applied to a pair of convex polytopes is described by the following procedure:

1. Initially, with $k = 0$, $\mathbf{v}_0$ is set to a random vector of non-zero magnitude.
2. While $\|\mathbf{v}_k\| \neq 0$, the algorithm loops through the following steps:
   a) Find $\mathbf{w}_k = s_{\mathcal{C}}(-\mathbf{v}_k)$ and add it to the set $\mathcal{W}_k$.
   b) If $\mathbf{v}_k \cdot \mathbf{w}_k \geq 0$, no collision is detected and the algorithm terminates (*i.e.* if point $\mathbf{w}_k$ does not go beyond the origin O in the $-\mathbf{v}_k$ direction, the polytope defined by $\mathcal{C}$ cannot envelop O and thus the pair of convex polytopes are not intersecting).
   c) Set $k = k + 1$, then set $\mathbf{v}_k$ as the point within $\mathcal{W}_{k-1}$ that is closest to O, *i.e.* $u(\mathcal{W}_{k-1})$.
   d) Set $\mathcal{W}_k$ as the smallest set of points found within $\mathcal{W}_{k-1}$, such that it includes $\mathbf{v}_k$. This ensures that $\mathcal{W}_k$ consists of $m + 1$ points or less at all times.
   e) Return to the beginning of the loop.
3. If $\|\mathbf{v}_k\| = 0$, a collision is detected and the algorithm may stop.

An example of this enhanced GJK algorithm can be seen in Figure 7.2, which depicts the Minkowski sum of the pair of convex polytopes represented in Figure 7.1(a). Initially, as seen in Figure 7.2(a), a search direction of $\mathbf{v}_0$ yields the point W_0. Since W_0 is located beyond O in the $-\mathbf{v}_0$ direction (as emphasized by the dashed line, which represents a line normal to $-\mathbf{v}_0$ passing through point W_0), the algorithm carries on through the next iteration. In Figure 7.2(b), setting $\mathbf{v}_1 = \mathbf{w}_0$ allows for point W_1 to be found and be added to $\mathcal{W}_1$. Once again W_1 is located beyond O in the $-\mathbf{v}_1$ direction and the algorithm carries on. In the following iteration, as depicted by Figure 7.2(c), $\mathbf{v}_2$ is set to equal to a point along the line between points W_0 and W_1, *i.e.* $\mathbf{v}_2 = u(\mathcal{W}_1)$, which is then used to obtain W_2. In this instance however, W_2 is not found to be located beyond O in the $-\mathbf{v}_2$ direction

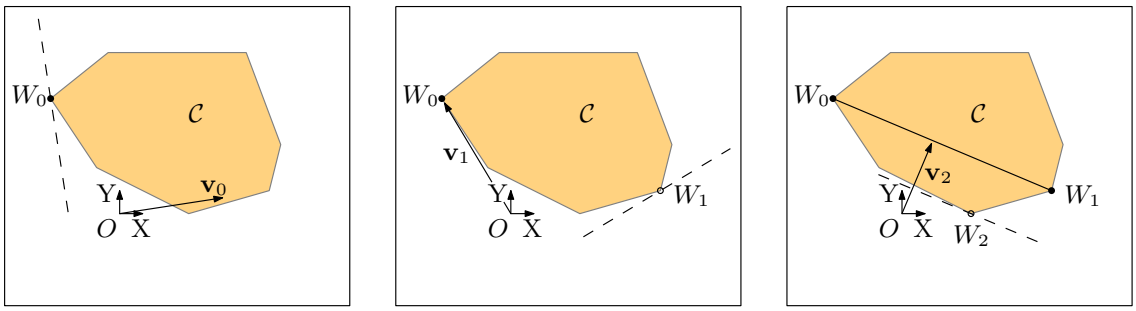


Figure 7.2: Three iterations of the enhanced GJK algorithm process for polytopes defined by sets of points $\mathcal{A}$ and $\mathcal{B}$ (from Figure 7.1(a)) visualized using the resulting Minkowski sum $\mathcal{C}$. (a) $k = 0$, (b) $k = 1$ and (c) $k = 2$.

and therefore the algorithm may terminate since no collision has occurred. For more information regarding this algorithm and the original GJK algorithm, the reader may refer to [57,58]. With the enhanced GJK algorithm having been defined, its use in the development of a collision avoidance strategy will be described in Section 7.3. However, prior to this, all objects found within a semi-automated rockbreaker system must be modeled as convex objects (this includes the rockbreaker's own links as well as all objects within its surrounding environment). Note that the collision detection between pairs of objects is only as accurate as the modeling accuracy of all collidable objects.

7.2 Modeling of the rockbreaker and its environment

In order to implement the enhanced GJK algorithm, all p collidable objects found within the working environment (*i.e.* the rigid bodies which constitute the rockbreaker and the objects located in its surrounding environment) must be modeled as convex objects. To model these objects, convex polytopes defined using sets of points labeled as $\mathcal{D}_h$ $(h = 1, 2, \ldots, p)$ are to be used. The polytope defined by $\mathcal{D}_h$ is defined by a set of v vertices whose position vectors are denoted as $\mathbf{d}_{hj}$ $(j = 1, 2, \ldots, v)$. Note that, for modeling simplicity and computational efficiency, $v \leq 8$ is imposed for each of the p polytopes. When modeling any object, the resulting polytope defined by $\mathcal{D}_h$ essentially represents an oversized envelope, completely encompassing the object (*i.e.* as a bounding box). This has the added benefit of providing an inherent factor of safety in the collision detection algorithm that may account for small deviations in the prescribed trajectories during the rockbreaker's motion. Moreover, this factor of safety may be modified at will simply by increasing the size of the polytopes

based on a scaling parameter.

Using the CAD model of a virtual rockbreaker system, sets of vertices $\mathcal{D}_h$ representing convex polytopes may be generated for each of the rockbreaker's first three links ($h = 1, 2, 3$). In doing so, the positions of the h^{th} link's vertices are initially defined relative the $(h+1)^{\text{th}}$ reference frame, *i.e.* $[\mathbf{d}_{hj}]_{h+1}$. In order to better represent its shape, the breaker link, for its part, is modeled as two separate convex polytopes (one for the breaker housing and another for the breaker tip). These may be represented by the sets of points $\mathcal{D}_4$ and $\mathcal{D}_5$, respectively, defined with respect to the $X_5Y_5Z_5$ reference frame ($[\mathbf{d}_{4j}]_5$ and $[\mathbf{d}_{5j}]_5$).

While each of the convex polytopes representing the rockbreaker's links were conveniently defined in reference frames attached to the links, the enhanced GJK algorithm requires that they be converted to a common reference frame. In this instance, the rockbreaker's base reference frame $X_1Y_1Z_1$ is used. Using Eqs. (3.3) and (3.4), the sets of vertices $\mathbf{d}_{hj}$ representing the h^{th} rockbreaker link, can be defined in the based frame as:

$$[\mathbf{d}_{hj}]_1 = [\mathbf{a}_1]_1 + \mathbf{R}_2^1[\mathbf{a}_2]_2 + \cdots + \mathbf{R}_2^1\mathbf{R}_3^2 \ldots \mathbf{R}_h^{h-1}[\mathbf{a}_h]_h + \mathbf{R}_2^1\mathbf{R}_3^2 \ldots \mathbf{R}_{h+1}^h[\mathbf{d}_{hj}]_{h+1} \tag{7.7}$$

with the exception of the convex object for the breaker tip, defined by $\mathcal{D}_5$, whose vertices can be described in the base frame as:

$$[\mathbf{d}_{5j}]_1 = [\mathbf{a}_1]_1 + \mathbf{R}_2^1[\mathbf{a}_2]_2 + \cdots + \mathbf{R}_2^1\mathbf{R}_3^2\mathbf{R}_4^3[\mathbf{a}_4]_4 + \mathbf{R}_2^1\mathbf{R}_3^2 \ldots \mathbf{R}_5^4[\mathbf{d}_{5j}]_5 \tag{7.8}$$

The remaining objects needed to be modeled as collidable objects are the static obstacles and zoning elements found within the rockbreaker's environment. Their models are defined as a set of points $\mathcal{D}_h$ (where $h = 6, 7, \cdots, p$), whose vertices $[\mathbf{d}_{hj}]_1$ are easily defined within the base reference frame $X_1Y_1Z_1$ due to their static nature. In order to obtain vertex positions relative the $X_1Y_1Z_1$ frame, a three-dimensional scanner could be used to measure the overall dimensions of the rockbreaker's surrounding environment, where the generated point cloud is then used to identify all $[\mathbf{d}_{hj}]_1$ needed to define all the convex polytopes defined by sets of points $\mathcal{D}_h$. For more complex geometries (such as concave wall structures found in excavated chambers in underground mines, as seen in Figure 7.3(a)), the concavities may be walled-off using smooth planes, as shown in Figure 7.3(b).

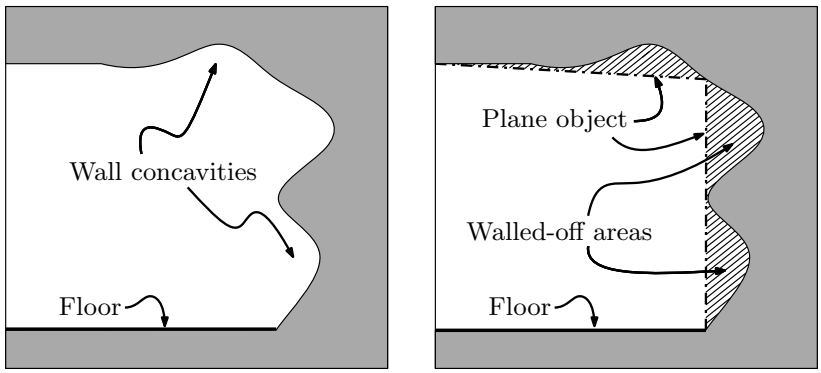


Figure 7.3: Modeling wall concavities within the rockbreaker's environment. (a) Wall concavities found within the work environment. (b) Plane objects used to delimit said concavities.

In this work, the modeling of the rockbreaker's surrounding environment is based on sample installation drawings/plans of excavated areas expected to house a rockbreaker system. From these installation drawings, all sets of points $\mathcal{D}_h$ required to define the convex polytopes representing the various static objects found within the work environment may be derived (at the designer's discretion). Furthermore, it should be noted that self-collisions between the breaker tip and the rockbreaker's base (which hasn't been modeled) are known to be feasible when the breaker tip is located in a position where $x \leq 0$. In order to be able to detect such a collision, a plane object (coincident with the Y_1Z_1 plane) can be used to represent the rockbreaker's base and be treated as a static collidable object within the work environment.

With all objects modeled (as shown in Figure 7.4), it is necessary to identify collision pairs (pairs of objects which may interact with one another) in order to avoid unnecessary collision checks between object pairs which are physically incapable of interaction. To identify these collision pairs, a strictly upper triangular binary matrix $\mathbf{H}$ of size $5 \times p$ is used as a lookup table. Manually setting the element of $\mathbf{H}$ located on its I^{th} row and J^{th} column, *i.e.* $H_{I,J}$, to 1 (*i.e.* "true") indicates a potential for the I^{th} convex object (*i.e.* a link of the rockbreaker) to collide with the J^{th} convex object (*i.e.* a rockbreaker link or an obstacle in the work environment). This method is implemented to ensure that self-collisions can be verified. When running a collision check for a specific rockbreaker configuration, the convex objects representing the rockbreaker's links must first be represented in the $X_1Y_1Z_1$ frame, using Eqs. (7.7) and (7.8), then the collision pairs, identified using $\mathbf{H}$, are checked for collision using

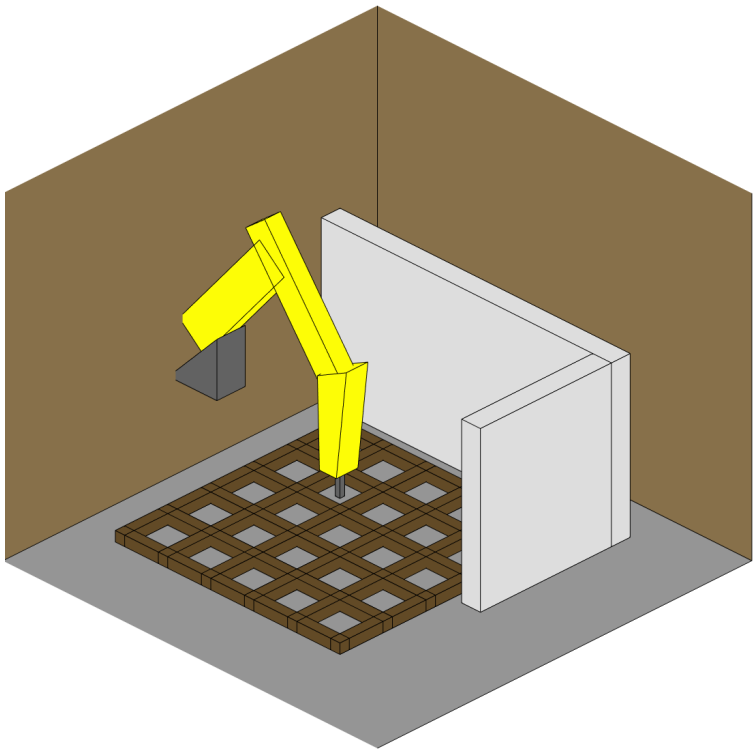


Figure 7.4: Visualization of a virtual rockbreaker system (configuration specific) and its environment represented as convex objects.

the enhanced GJK algorithm. Although physics engines and collision detection checks for complex systems (*i.e.* which have a great number of collidable objects) usually take a two-stage approach for collision detection, consisting of a broad phase for collision pair identification and a narrow phase for collision detection [51, 52, 60], this approach is deemed unnecessary for the rockbreaker system due to the low number of collidable objects expected to be found within the work environment. Lastly, although the prismatic hydraulic actuators found on the rockbreaker aren't currently modeled, these may be included if deemed necessary, modeling them as two separate collidable objects, in a similar fashion as the rockbreaker's links. Their vertices $\mathbf{d}_{hj}$ would be described with respect to fixed reference frames found on the actuators' pins and would need to be interpreted in the base frame using a similar approach as the one introduced in Eq. (7.7).

7.3 Collision detection strategy

Since collision detection is accomplished using the enhanced GJK algorithm, the rockbreaker's configuration for any motion must be known to ensure the modeled rockbreaker objects are properly represented for the collision detection. If provided with entire rockbreaker trajectories, the resulting collision detection algorithm can be used to verify that they are collision-free. These trajectories may then be provided to a controller to run the rockbreaker.

For the purposes of generating collision-free trajectories, two path planning approaches are considered. The first path planning algorithm that is considered generates trajectories with decoupled swing joint motions, similar to the path planner presented in Chapter 5. However, this algorithm is modified such that it implements continuous flow rate scaling as was proposed in Chapter 6. The second approach for path planning is based on the Cartesian straight-line motion planner provided in Chapter 6. Since both path planning approaches utilize the same time parameterization technique (*i.e.* continuous flow rate scaling), the resulting trajectories are expected to be time optimal given the types of paths that are imposed.

In terms of ideal trajectory planning methodology, both the aforementioned approaches present benefits and challenges depending on the circumstances. For instance, the decoupled swing joint motion planner guarantees the resulting trajectory remains within the workspace of the rockbreaker despite its non-convexity (as long as the motions are found within the work plane where $x \geq 0$). Furthermore, the resulting trajectory is time optimal as it avoids unnecessary hydraulic actuator motions (*i.e.* extension followed by contraction). However, the decoupled approach does not provided a fully predictable path in Cartesian space as the swing and planar arm motions are independent of one another. In particular, the breaker tip is not guaranteed to remain within the region located above the grizzly during a given motion, which could lead to collisions with static obstacles surrounding the grizzly. This is illustrated in Figure 7.5 for an example scenario consisting of a motion from an initial position $\mathbf{p}_I = [1.25, 2.75, -0.5]^T$ to a final position $\mathbf{p}_F = [6.75, -2.75, -0.5]^T$. On the other hand, the Cartesian straight-line approach guarantees that the breaker tip remains above the grizzly during any motions within the work plane (see Figure 7.5). There are associated drawbacks with this method, however, notably that the resulting trajectories may lead to unnecessary motions of

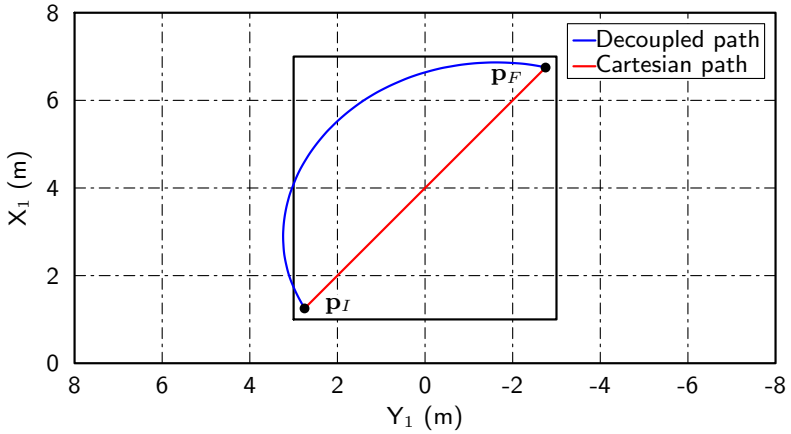


Figure 7.5: Time-optimal paths obtained for the example scenario using the decoupled swing joint motion approach and the Cartesian straight-line approach.

the hydraulic actuators (*e.g.* extension followed by contraction) and it does not guarantee that the breaker tip remains within the rockbreaker's non-convex workspace.

In order to provide a collision-free trajectory to the controller, the following path planning strategy is employed:

- Initially, a trajectory generated using the decoupled swing joint motion approach is verified for potential collisions. If the resulting trajectory satisfies this criteria, it is supplied to the controller.
- If the aforementioned trajectory fails to be collision-free, a trajectory is generated using the Cartesian straight-line approach and verified for collisions.
- If both approaches fail to identify a collision-free trajectory, a new path planning approach must be developed.

Since instances where both path planning approaches fail to generate a feasible trajectory exist (*i.e.* paths leading to collision and/or exiting the workspace), a third path planning approach must be introduced. This new (hybrid) approach should ensure that the breaker tip remains within the workspace and above the grizzly for the entirety of the trajectory, as this will significantly reduce the chances of collisions. As such, this hybrid approach will be derived from a modified version of decoupled swing joint motion path planner which also ensures the breaker tip remains above the grizzly at all times.

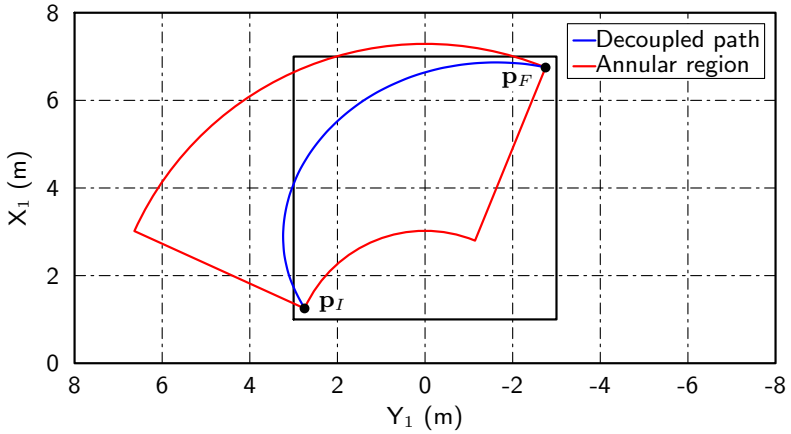


Figure 7.6: Annular region based on the decoupled swing joint motion trajectory for the example scenario.

The hybrid approach begins with the boundary identification of the area defined by the grizzly as line segments ($\mathcal{R}_g$) as well as the boundaries of an annular region, which is defined using a combination of lines segments and circular arcs ($\mathcal{R}_a$). The annular region is defined as the area which encloses the decoupled swing joint motion. It is known that this resulting motion is the combination of an angular motion about the swing joint and a radial motion (straight line motion along the X_2 axis). For any generated path, the radial and angular motions are known to be bounded, where the minimum and maximum radial distances with respect to $X_1Y_1Z_1$ are known as well as θ_I and θ_F. With these bounds, the annular region which encloses the decoupled swing joint path may be determined (as shown in Figure 7.6). Next, the boundaries of the admissible region (*i.e.* region which remains above the grizzly) are identified by calculating the intersections between the grizzly area and the annular region (*i.e.* intersections between $\mathcal{R}_g$ and $\mathcal{R}_a$). The resulting boundary consists of a set of line segments and circular arcs ($\mathcal{P}$). From this set, all possible alternative paths along the admissible region's boundary which connect the initial and final rockbreaker poses are identified. For the example scenario illustrated in Figures 7.5 and 7.6, this results in two alternative paths, $\mathcal{P}_1$ and $\mathcal{P}_2$, as shown in Figure 7.7. The shortest alternative path (in terms of the number of line/arc segments) is chosen and trajectories are generated for each segment using the decoupled swing joint motion approach for circular arc segments and the Cartesian straight-line approach for line segments. Note that, in the case where both alternative paths are of identical length, the first alternative path which was identified is to be used. These trajectories are then compounded into a single trajectory where

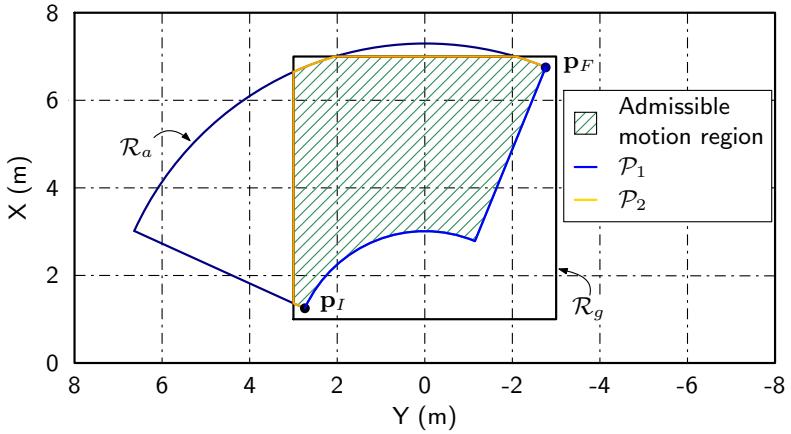


Figure 7.7: Admissible motion region bounded by $\mathcal{P}$ and alternative paths $\mathcal{P}_1$ and $\mathcal{P}_2$ for the example scenario. Note that the paths are labeled according to the path planning approach, where DA, CA and HA denote the paths generated by the decoupled swing joint motion, Cartesian straight-line motion and hybrid approach, respectively.

the rockbreaker's motion stops momentarily during the transition between the separate trajectory segments. Furthermore, it should be noted that line segments may be approximated as circular arcs of very large radius to simplify the process (*i.e.* eliminating the need to deal with two types of geometrical entities) if desired. Once a trajectory consisting of multiple segments has been assembled, each of its segments is verified for collisions. If collisions are found, trajectories are generated for the next shortest alternative path and they are verified for possible collisions. In the instance that no trajectory is collision-free, the system will need to be operated manually by the operator. However, those instances are expected to be extremely rare and none have been observed throughout the algorithm validation process. Moreover, it is expected that such cases would be the result of poor installation design of the rockbreaker system. A summary of the trajectory planning with collision avoidance methodology using the three various path planning approaches is provided in the form of a flow chart shown in Figure 7.8. Furthermore, the trajectories produced by each approach for three example scenarios can be seen in Figure 7.9, where it has been considered that any departure from the grizzly area results in a collision. It should be noted that when inspecting the trajectories using the enhanced GJK algorithm, the verification is only performed for discrete time steps along the trajectory. Although undetected collisions could still occur between consecutive time steps, this risk is mitigated by the rockbreaker's relatively slow speed, the use of slightly larger than required convex polytopes to model its links and the setting of the GJK's algorithm's frequency of execution

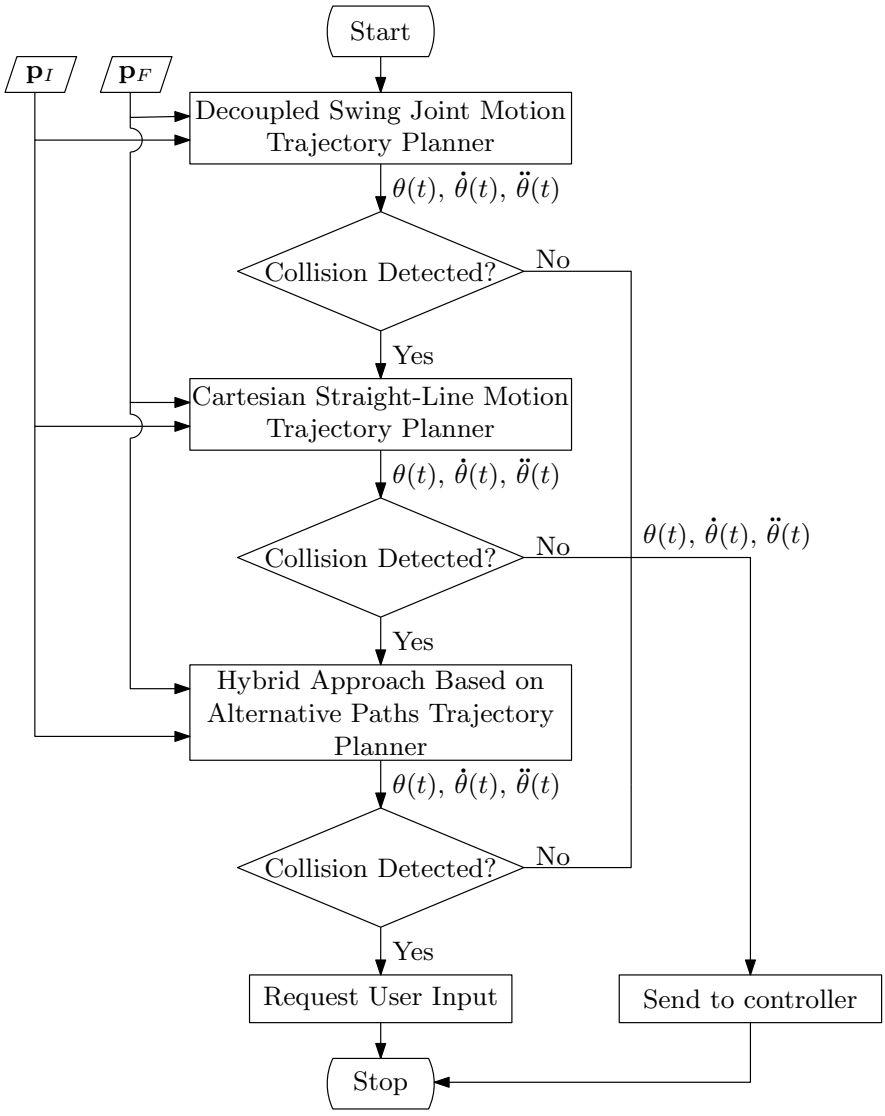


Figure 7.8: Summary of the trajectory planning with collision avoidance approach given $\mathbf{p}_I$ and $\mathbf{p}_F$.

to be sufficiently high (but low enough to maintain acceptable computation time, *e.g.* 50 Hz).

7.4 Results

The results that follow were obtained based on a virtual rockbreaker system having the kinematic parameter values listed in Appendix B. The rockbreaker system and its surrounding virtual environment were modeled as convex polytopes which were previously shown in Figure 7.4. This led to the creation of a total of 22 convex objects (5 for the rockbreaker and 17 for the surrounding environment). A collision pair matrix $\mathbf{H}_{5\times 22}$ was then generated to identify the pairs of objects that

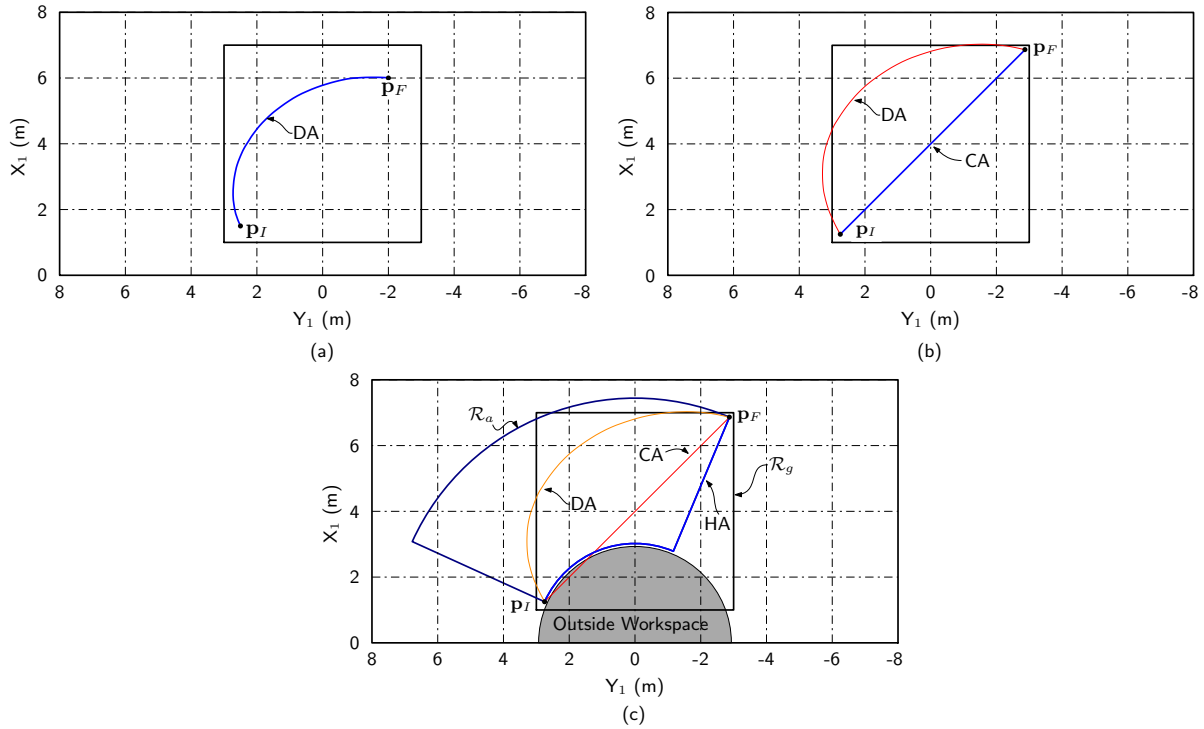


Figure 7.9: Trajectory planning with collision avoidance from $\mathbf{p}_I$ to $\mathbf{p}_F$ using: (a) the decoupled swing joint motion approach (DA), (b) the Cartesian straight-line motion approach (CA) and (c) the hybrid approach based on alternative paths (HA).

could collide. Three different trajectories were computed using the collision-free trajectory planning algorithm, showcasing all three possible outcomes using the methodology described in this work. The initial and final positions ($\mathbf{p}_I$ and $\mathbf{p}_F$) for theses trajectories are as follows:

$$\mathbf{p}_{I_1} = [1.1,\ \text{-}2.9,\ \text{-}1]^T\ \text{m} \qquad \mathbf{p}_{F_1} = [6.9,\ 2.9,\ \text{-}1]^T\ \text{m}$$

$$\mathbf{p}_{I_2} = [1.1,\ 2.9,\ \text{-}1]^T\ \text{m} \qquad \mathbf{p}_{F_2} = [6.9,\ 2.9,\ \text{-}1]^T\ \text{m}$$

$$\mathbf{p}_{I_3} = [1.0,\ 3.0,\ 1]^T\ \text{m} \qquad \mathbf{p}_{F_3} = [6.5,\ \text{-}2.5,\ 1]^T\ \text{m}$$

The resulting collision-free trajectories are shown in Figure 7.10. The total run time of 10 iterations for each trajectory computation scenario was, on average of 4.79, 9.02 and 8.98 seconds, respectively, whose trajectories are defined using discrete time steps taken at a sampling frequency of 1 kHz and with collision checks computed at a frequency of 50 Hz[1]. Although the computation times may seem excessive, the code utilized to generate these trajectories suffers from MATLAB overhead during run

[1]Simulations done using a Lenovo ThinkPad W550s with a Intel i7-5500U CPU @2.4 GHz, 8 GB RAM

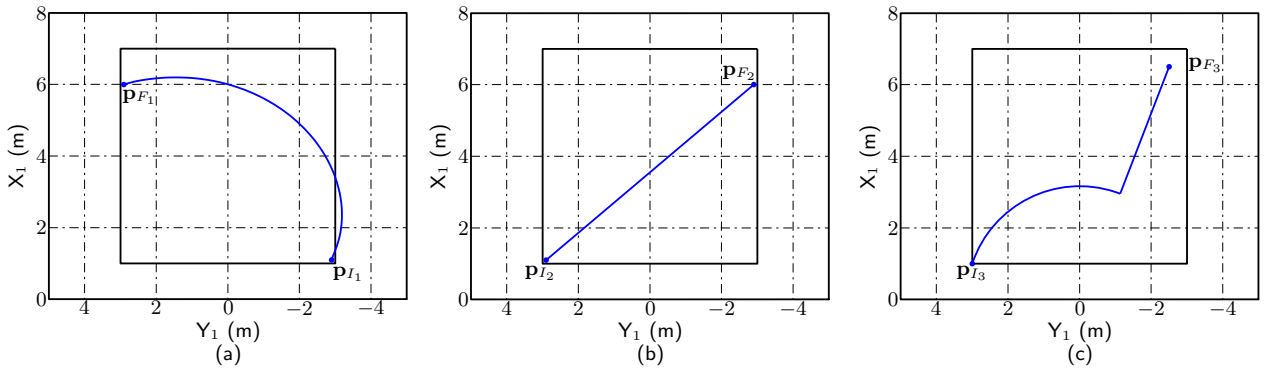


Figure 7.10: Possible outcomes of trajectories generated using the trajectory planning algorithm with collision avoidance. Example of (a) decoupled trajectory, (b) Cartesian trajectory and (c) hybrid trajectory.

time as well as some sub-optimal subroutines which could benefit from compilation into C-code. Furthermore, the enhanced GJK algorithm used during these simulations was first compiled as a .mex file using C-code and performed a collision check between a pair of objects within $\sim 10-16$ μs. Alterations to the original trajectory planning algorithms were made to allow for compilation of the code (where the optimization routine for the ramping segments of the trajectories were computed using a sequential linear programming routine rather than the sequential quadratic programming routine from `fmincon`), where the resulting average run time for each trajectory computation iteration (out of 100) was of 0.09, 0.18 and 0.20 seconds for the three cases, respectively. Lastly, although collision checks were performed between various combinations of all objects seen in Figure 7.4, the number of collision checks could have been reduced since some collisions are known to be infeasible when the breaker tip is found near/within the work plane (*e.g.* the breaker tip cannot collide with the grizzly's modeled grid mesh or the walled-off area defined by the plane to the left of the rockbreaker), which would reduce computation time for collision detection.

7.5 Conclusion

The use of the collision-free trajectory planning algorithm for a semi-autonomous rockbreaker system has been introduced. Although the algorithm does not absolutely guarantee a collision-free trajectory will be found (due to verifications being made at finite time intervals), cases where the algorithm was not successful have yet to be identified. Furthermore, most encountered cases only went as far

as generating trajectories using the Cartesian straight-line motion planner whilst the hybrid motion planner was rarely needed. The algorithm has proven to be robust in supplying feasible trajectories during various simulations, while also being adaptable for various rockbreaker installation designs. Future work will entail the incorporation of a broad phase for collision detection for more complex environments (*i.e.* a preliminary collision detection approach to identify potential collision pairs) and more detailed object models which would effectively remove the need for $\mathbf{H}$, adding coherence to the enhanced GJK algorithm (selectively choosing a search direction between collision pairs) and exploring the use of geometric primitives to define the convex objects within the system.

Chapter 8

Conclusion

In conclusion, the development of collision-free trajectory planning algorithms necessary in order to semi-automate a commercial rockbreaker system was achieved. The benefits of such a system are numerous, notably an increase in operation efficiency, productivity and workplace safety while reducing operating and maintenance costs. Summaries of the resulting algorithm development and results are provided in Sections 8.1, 8.2 and 8.3, regarding the dynamic modeling, trajectory planning and collision avoidance strategy for a rockbreaker system.

8.1 Dynamic modeling

A dynamic model for the rockbreaker system was developed using the Lagrange-Euler approach. The full model of the system considered the dynamic effects of both the links and hydraulic actuators on the torques required at each of its revolute joints. From this model, simplifications were explored to reduce the complexity of the dynamic model while providing a good approximation of the system. From dynamic simulation results, it was found that the Coriolis and centrifugal effects had little impact on the torques felt at the rockbreaker's revolute joints. The gravity term was found to be the main contributor, accounting for up to 97.5% of the joint torque in some instances. Furthermore, the influence of the entire dynamic effects of the hydraulic actuators on the joint torques accounted for up to 10% of the total joint torque in some instances, which is significant and should be considered when modeling such systems. The most promising of the simplified models was one that accounted

for the generalized inertia term for the rockbreaker's link and the gravity term of the entire system. However, these models are not without their limitations, including their omission of friction terms (*i.e.* Coulomb and viscous friction) which are difficult to model yet have significant influence on such systems. Furthermore, the resulting dynamic models are subject to errors, wherein parameters values used in the simulations may differ from a real system. These issues may be investigated in future work, when a commercial rockbreaker system is available for dynamic testing. Furthermore, a comparison of the system's performance during semi-autonomous control utilizing model based and non-model based controllers could offer further insight on the benefits and/or necessity of a dynamic model for this system.

8.2 Trajectory planning

Robust trajectory planning algorithms that generate time-optimal motions for the rockbreaker system (given the types of paths that were imposed) were presented in Chapters 5 and 6. Two approaches to path planning were introduced: decoupled swing joint motion and Cartesian straight-line motion. The resulting trajectories ensured flow rate limitations of the system were respected while motions remained smooth and continuous. This was accomplished through the scaling of flow rates, initially using a constant scaling factor for the entire trajectory (as was the case in Chapter 5) followed by the use of a scaling ratio function, which provided an optimal scaling factor for each discrete time step of the trajectory (as described in Chapter 6). However, the approaches taken by these algorithms only provided offline trajectories which cannot be altered while the rockbreaker is in motion. Furthermore, if one were to compound multiple generated trajectories (similar to creating a path along via points), the resulting trajectory would have introduced inefficiencies arising from intermediary ramping motion segments. Future work to mitigate these shortcomings would entail rendering the presented algorithms into online trajectories, which ensure smooth and continuous motions.

8.3 Collision avoidance

A strategy which ensures collision-free trajectories within a static environment was introduced in Chapter 7. The approach consisted of validating generated trajectories for potential collisions using

an enhanced GJK algorithm. If trajectories using the decoupled swing joint motion and Cartesian straight-line motion resulted in collisions (although rare in practice), a hybrid trajectory planning algorithm, which ensured the breaker tip remained within the rockbreaker's workspace and above the grizzly, was used. While it remains possible for all three of these trajectory planning algorithms to fail to generate collision-free trajectories, such a case has not yet been encountered.

While simulation results using the collision-free trajectory planning algorithms yielded positive results, the performance of the proposed approach is dependent on the modeling accuracy of collidable objects. Furthermore, if deviations of the rockbreaker from its prescribed path during its motion are significantly high due to poor control performance, the resulting motions could also still lead to collisions. To mitigate these shortcomings, further research on the possibility of using online collision detection systems (*e.g.* vision-based) should be pursued. This would have the added benefit of being able to deal with dynamic environments .

References

[1] Louis-Francis Tremblay, Marc Arsenault, and Meysar Zeinali. Simplification of the dynamic model of a hydraulic rockbreaker for the purpose of implementation in a model-based control scheme. *Transactions of the Canadian Society of Mechanical Engineering (in press)*, 2018.

[2] Peter Corke, Jonathan Roberts, Jock Cunningham, and David Hainsworth. *Springer Handbook of Robotics*, chapter Mining Robotics, pages 1127–1150. Springer, 2008.

[3] Daniel Ortiz Morales, Simon Westerberg, Pedro X. La Hera, Uwe Mettin, Leonid Freidovich, and Anton S. Shiriaev. Increasing the level of automation in the forestry logging process with crane trajectory planning and control. *Journal of Field Robotics*, 31(3):343–363, 2014.

[4] Anders Hansson and Martin Servin. Semi-autonomous shared control of large-scale manipulator arms. *Control Engineering Practice*, 18(9):1069–1076, September 2010.

[5] B. J. Alshaer, T. T. Darabseh, and M. A. Alhanouti. Path planning, modeling and simulation of an autonomous articulated heavy construction machine performing a loading cycle. *Applied Mathematical Modelling*, 37(7):5315–5325, April 2013.

[6] C. Y. Ho and Yao Jianchi. Computer-controlled automated jumbo drilling robot manipulator. *International Journal of Mining and Geological Engineering*, 4(4):303–318, January 1986.

[7] Anna Gustafson. Automation of load haul dump machines. Luleå University of Technology, 2011.

[8] Jonathon C. Ralston, David C. Reid, Mark T. Dunn, and David W. Hainsworth. Longwall automation: Delivering enabling technology to achieve safer and more productive underground

mining. *International Journal of Mining Science and Technology*, 25(6):865–876, November 2015.

[9] Elliot S. Duff, Jonathan M. Roberts, and Peter I. Corke. Automation of an underground mining vehicle using reactive navigation and opportunistic localization. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, volume 4, pages 3775–3780, 2003.

[10] Jonathan M. Roberts, Elliot S. Duff, Peter I. Corke, Pavan Sikka, Graeme J. Winstanley, and Jock Cunningham. Autonomous control of underground mining vehicles using reactive navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 3790–3795, 2000.

[11] A. Boeing. A remotely operated robotic rock breaker with collision avoidance for the mining industry. In *Proceedings of the 30th International Symposium on Automation and Robotics in Construction and Mining*, pages 875–884, Montreal, QC, Canada, 2013. Canadian Institute of Mining, Metallurgy and Petroleum.

[12] David A. Bradley and Derek W. Seward. Development, control and operation of an autonomous robotic excavator. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 21(1):73 – 97, 1998.

[13] Pyung Hun Chang and Soo-Jin Lee. A straight-line motion tracking control of hydraulic excavator system. *Mechatronics*, 12(1):119–138, 2002.

[14] Anthony Stentz, John Bares, Sanjiv Singh, and Patrick Rowe. A robotic excavator for autonomous truck loading. *Autonomous Robots*, 7(2):175–186, Sep 1999.

[15] H.N. Cannon. Extended earthmoving with an autonomous excavator. Master's thesis, The Robotics Institute, Carnegie Mellon University, 1999.

[16] Vivek Chacko, Hongnian Yu, Shuang Cang, and Luige Vladareanu. State of the art in excavators. In *Proceedings of the IEEE International Conference on Advanced Mechatronic Systems*, pages 481 – 488, Kumamoto, Japan, 2014.

[17] Elliot Duff, Con Caris, Adrian Bonchis, Ken Taylor, Chris Gunn, and Matt Adcock. The development of a telerobotic rock breaker. *Springer Tracts in Advanced Robotics*, 62:411–420, 2010.

[18] Murali Krishna. Optimal motion generation for hydraulic robots. Technical Report CMU-RI-TR-98-09, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, March 1998.

[19] Bin Zhang, Shuang Wang, Yuting Liu, and Huayong Yang. Research on trajectory planning and autodig of hydraulic excavator. *Mathematical Problems in Engineering*, May 2017.

[20] Quang Ha, Miguel Santos, Quang Nguyen, David Rye, and Hugh Durrant-Whyte. Robotic excavation in construction automation. *IEEE Robotics and Automation Magazine*, 9(1):20–28, 2002.

[21] Daniel Schmidt, Fabian Gockel, and Karsten Berns. Behavior-based collision avoidance using a cylinder-coordinate octree. volume 531, pages 143 – 155, Shanghai, China, 2017.

[22] Gregor Zolynski, Daniel Schmidt, and Karsten Berns. Safety for an autonomous bucket excavator during typical landscaping tasks. volume 20, pages 357 – 368, Cluj-Napoca, Romania, 2014.

[23] Chris Leger, Patrick Rowe, John Bares, Scott Boehmke, and Anthony Stentz. Obstacle detection and safeguarding for a high-speed autonomous hydraulic excavator. *Proceedings of SPIE - The International Society for Optical Engineering*, 3525:146 – 156, 1998.

[24] Michael Greenspan, John Ballantyne, and Mike Lipsett. Sticky and slippery collision avoidance for tele-excavation. volume 3, pages 1666 – 1671, Grenoble, Fr, 1997.

[25] S. Singh. State of the art in automation of earthmoving. *Journal of Aerospace Engineering*, 10(4):179–188, 1997.

[26] Sanjiv Singh. Synthesis of tactical plans for robotic excavation. pages 185 –, 1995.

[27] Jiaqi Xu and Hwan-Sik Yoon. A review on mechanical and hydraulic system modeling of excavator manipulator system. *Journal of Construction Engineering*, pages 1 – 11, 2016.

[28] Nikolaos Mavridis, Georgios Pierris, Paolo Gallina, Nikolaos Moustakas, and Alexandros Astaras. Subjective difficulty and indicators of performance of joystick-based robot arm teleoperation with auditory feedback. In *Proceedings of the 17th International Conference on Advanced Robotics*, pages 91–98, 2015.

[29] Leonhard E. Bernold. Quantitative assessment of backhoe operator skill. *Journal of Construction Engineering and Management*, 133(11):889 – 899, 2007.

[30] Jungwon Yoon and Auralius Manurung. Development of an intuitive user interface for a hydraulic backhoe. *Automation in Construction*, 19(6):779 – 790, 2010.

[31] Enrique Cabello, M. Araceli Sanchez, and Javier Delgado. A new approach to identify big rocks with applications to the mining industry. *Real-Time Imaging*, 8(1):1–9, 2002.

[32] R. A. Salinas, U. Raff, and C. Farfan. Automated estimation of rock fragment distributions using computer vision and its application in mining. *IEE Proceedings: Vision, Image and Signal Processing*, 152(1):1–8, 2005.

[33] L. Sciavicco and B. Siciliano. *Modelling and Control of Robot Manipulators*. Springer, 2nd edition, 2000.

[34] J.-J. E. Slotine and W. Li. On the adaptive control of robot manipulators. *The International Journal of Robotics Research*, 6(3):49–59, 1987.

[35] J. J. Craig, P. Hsu, and S. S. Sastry. Adaptive control of mechanical manipulators. *The International Journal of Robotics Research*, 6(2):16–28, 1987.

[36] P. K. Vaha and M. J. Skibniewski. Dynamic model of excavator. *Journal of Aerospace Engineering*, 6(2):148–158, 1993.

[37] A. J. Koivo, M. Thoma, E. Kocaoglan, and J. Andrade-Cetto. Modeling and control of excavator dynamics during digging operation. *Journal of Aerospace Engineering*, 9(1):10–18, 1996.

[38] S. Frimpong, Y. Hu, and H. Inyang. Dynamic modeling of hydraulic shovel excavators for geomaterials. *Internation Journal of Geomechanics*, 8(1):20–29, 2008.

[39] S. Salinic, G. Boskovic, and M. Nikolic. Dynamic modelling of hydraulic excavator motion using Kane's equations. *Automation in Construction*, 44(1):56–62, 2014.

[40] B. P. Patel and J.M. Prajapati. Dynamics of mini hydraulic backhoe excavator:a lagrange-euler (l-e) approach. *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, 8(1):195–204, 2014.

[41] Young Bum Kim, Hyuk Kang, Jun Hyeong Ha, Moo Seung Kim, Pan Young Kim, Ssang Jae Baek, and Jinsoo Park. A study on the virtual digging simulation of a hydraulic excavator. In *Proceedings of the 28th International Symposium on Automation and Robotics in Construction*, pages 95–100. IIT Madras, 2011.

[42] Young Bum Kim, Junhyoung Ha, Hyuk Kang, Pan Young Kim, Jinsoo Park, and F. C. Park. Dynamically optimal trajectories for earthmoving excavators. *Automation in Construction*, 35:568–578, 2013.

[43] Murali Krishna and John Bares. Constructing hydraulic robot models using memory-based learning. *Journal of Aerospace Engineering*, 12(2):34 – 42, 1999.

[44] Bongju Lee and H. Jin Kim. Trajectory generation for an automated excavator. In *Proceedings of the IEEE International Conference on Control, Automation and Systems*, pages 716–719, 2014.

[45] Onesmus M. Muvengei. Simulation of the dynamic behavior of an excavator due to the interacting mechanical and hydraulic dynamics. Master's thesis, Jomo Kenyatta University of Agriculture and Technology, 2008.

[46] Hui Shao, Hiroshi Yamamoto, Yuki Sakaida, Takashi Yamaguchi, Yuji Yanagisawa, and Akira Nozue. Automatic excavation planning of hydraulic excavator. In *Proceedings of the 1st International Conference on Intelligent Robotics and Applications*, pages 1201–1211. Springer, 2008.

[47] J. Jeevamalar and S. Ramabalan. Optimal trajectory planning for autonomous robots - A review. In *Proceedings of the IEEE International Conference on Advances in Engineering, Science and Management*, pages 269–275, Nagapattinam, Tamil Nadu, India, 2012.

[48] Mariana Ratiu and Mariana Adriana Prichici. Industrial robot trajectory optimization - A review. In *MATEC Web of Conferences*, volume 126, Baile Felix, Romania, May 2017. EDP Sciences.

[49] Anthony Esposito. *Fluid Power with Applications*. Pearson Education Inc., 7th edition, 2009.

[50] Derek Seward and Frank Margrave. Lucie the robot excavator-design for system safety. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 963 – 968, April 1996.

[51] Ian Robert Ballantyne and Paul Kelly. Collision overload: Reducing the impact in real-time physics. Master's thesis, Imperial College, 2007.

[52] Brian Mirtich. Efficient algorithms for two-phase collision detection. In *Practical Motion Planning in Robotics*, pages 203–223. John Wiley & Sons, 1998.

[53] Gino Van Den Bergen. Proximity queries and penetration depth computation on 3d game objects. In *Game developers conference*, volume 170, 2001.

[54] S. Gottschalk, M.C. Lin, and D. Manocha. Obbtree: A hierarchical structure for rapid interference detection. pages 171 – 180, New Orleans, LA, USA, 1996. Oriented bounding box trees (OBBTrees);Rapid interference detection;.

[55] Ming C. Lin and John F. Canny. A fast algorithm for incremental distance calculation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1008 – 1014, 1991.

[56] Ming C. Lin. *Efficient Collision Detection for Animation and Robotics*. PhD thesis, Department of Electrical Engineering and Computer Science, University of California at Berkeley, 1993.

[57] Gino Van den Bergen. A fast and robust gjk implementation for collision detection of convex objects. *Journal of Graphics Tools*, 4(2):7–25, 1999.

[58] Elmer G. Gilbert, Daniel W. Johnson, and S.Sathiya Keerthi. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE journal of robotics and automation*, 4(2):193 – 203, 1988.

[59] Stephen Cameron. Enhancing gjk: Computing minimum and penetration distances between convex polyhedra. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 3112 – 3117, 1997.

[60] Erwin Coumans. *Bullet 2.82 Physics SDK Manual*, 2013.

[61] Jonathan D Cohen, Ming C Lin, Dinesh Manocha, and Madhav Ponamgi. I-collide: An interactive and exact collision detection system for large-scale environments. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 189–ff. ACM, 1995.

[62] Kevin M. Lynch and Frank C. Park. *Modern Robotics: Mechanics, Planning and Control.* Cambridge University Press, 2017.

[63] Erwin Coumans. Opencl game physics. Presentation, November 2009.

[64] Jorge Angeles. *Fundamentals of robotic mechanical systems: Theory, methods, and algorithms.* Springer Science+Business Media, third edition, 2007.

[65] John J. Craig. *Introduction to Robotics: Mechanics and Control.* Pearson, third edition, 2004.

[66] P Kokotovic and R Marino. On vanishing stability regions in nonlinear systems with high-gain feedback. *IEEE Transactions on Automatic Control*, 31(10):967–970, 1986.

[67] Quang-Cuong Pham. A general, fast, and robust implementation of the time-optimal path parameterization algorithm. *IEEE Transactions on Robotics*, 30(6):1533 – 1540, 2014.

[68] Ralph Tyrell Rockafellar. *Convex analysis.* Princeton university press, 1970.

Appendix A

Parameter values used in dynamic simulations

Table A.1: Denavit-Hartenburg parameter values.

i	a_i (m)	b_i (m)	α_i (deg.)
1	0.349	1.049	90
2	3.397	0.000	0
3	2.642	0.000	0
4	2.219	0.000	0

Table A.2: Masses of links, actuator barrels and piston assemblies.

$$m_{l_1} = 1029.59 \text{ kg}, \quad m_{l_2} = 1313.70 \text{ kg}, \quad m_{l_3} = 975.41 \text{ kg}, \quad m_{l_4} = 2082.98 \text{ kg}$$
$$m_{b_0} = m_{b_1} = 51.61 \text{ kg}, \quad m_{b_2} = m_{b_3} = m_{b_4} = 134.96 \text{ kg}, \quad m_{p_0} = m_{p_1} = 33.52 \text{ kg}, \quad m_{p_2} = m_{p_3} = m_{p_4} = 84.99 \text{ kg}$$

Table A.3: Positions of links' centres of mass.

$$[\mathbf{s}_{l_1}]_2 = \begin{bmatrix} -0.044 \\ -0.432 \\ -0.006 \end{bmatrix} \text{ m}, \quad [\mathbf{s}_{l_2}]_3 = \begin{bmatrix} -1.618 \\ 0.044 \\ -0.001 \end{bmatrix} \text{ m}, \quad [\mathbf{s}_{l_3}]_4 = \begin{bmatrix} -1.865 \\ 0.082 \\ -0.001 \end{bmatrix} \text{ m}, \quad [\mathbf{s}_{l_4}]_5 = \begin{bmatrix} -1.001 \\ 0.126 \\ 0.002 \end{bmatrix} \text{ m}$$

Table A.4: Positions of actuator barrels' centres of mass.

$$[\mathbf{s}_{b_0}]'_0 = [\mathbf{s}_{b_1}]'_1 = \begin{bmatrix} 0.396 \\ 0.000 \\ 0.000 \end{bmatrix} \text{ m}, \quad [\mathbf{s}_{b_2}]'_2 = \begin{bmatrix} 0.703 \\ -0.005 \\ 0.001 \end{bmatrix} \text{ m}, \quad [\mathbf{s}_{b_3}]'_3 = [\mathbf{s}_{b_4}]'_4 = \begin{bmatrix} 0.703 \\ 0.005 \\ -0.001 \end{bmatrix} \text{ m}$$

Table A.5: Positions of actuator piston assemblies' centres of mass.

$$[\mathbf{s}_{p_0}]'_0 = [\mathbf{s}_{p_1}]'_1 = \begin{bmatrix} -0.394 \\ 0.000 \\ 0.000 \end{bmatrix} \text{ m}, \quad [\mathbf{s}_{p_2}]'_2 = [\mathbf{s}_{p_3}]'_3 = [\mathbf{s}_{p_4}]'_4 = \begin{bmatrix} -0.783 \\ 0.000 \\ 0.000 \end{bmatrix} \text{ m}$$

Table A.6: Positions of A_j nodes with respect to R_i nodes.

$$[\mathbf{u}_0]_1 = \begin{bmatrix} -1.163 \\ -0.375 \\ 0.419 \end{bmatrix} \text{ m}, \quad [\mathbf{u}_1]_1 = \begin{bmatrix} -1.163 \\ 0.375 \\ 0.419 \end{bmatrix} \text{ m}, \quad [\mathbf{u}_2]_2 = \begin{bmatrix} 0.318 \\ -0.706 \\ 0 \end{bmatrix} \text{ m}, \quad [\mathbf{u}_3]_3 = \begin{bmatrix} -2.254 \\ 0.434 \\ 0 \end{bmatrix} \text{ m}, \quad [\mathbf{u}_4]_4 = \begin{bmatrix} -2.316 \\ 0.405 \\ 0 \end{bmatrix} \text{ m}$$

Table A.7: Inertial matrices of links.

$$[\mathbf{I}_{l_1}]_2 = \begin{bmatrix} 158.98 & 0.94 & -0.63 \\ 0.94 & 65.74 & -1.79 \\ -0.63 & -1.79 & 164.75 \end{bmatrix} \text{ kg} \cdot \text{m}^2, \qquad [\mathbf{I}_{l_2}]_3 = \begin{bmatrix} 50.53 & 622.48 & -1.66 \\ 622.48 & 1079.78 & 0.03 \\ -1.66 & 0.03 & 1468.53 \end{bmatrix} \text{ kg} \cdot \text{m}^2$$

$$[\mathbf{I}_{l_3}]_4 = \begin{bmatrix} 797.82 & -489.64 & 0.66 \\ -489.64 & 347.07 & -0.11 \\ 0.66 & -0.11 & 1117.27 \end{bmatrix} \text{ kg} \cdot \text{m}^2, \quad [\mathbf{I}_{l_4}]_5 = \begin{bmatrix} 107.94 & 203.52 & -3.73 \\ 203.52 & 1355.62 & 0.68 \\ -3.73 & 0.68 & 1397.74 \end{bmatrix} \text{ kg} \cdot \text{m}^2$$

Table A.8: Inertial matrices of actuator barrels.

$$[\mathbf{I}_{b_0}]_{0'} = [\mathbf{I}_{b_1}]_{1'} = \begin{bmatrix} 0.22 & 0.00 & 0.00 \\ 0.00 & 3.70 & 0.00 \\ 0.00 & 0.00 & 3.71 \end{bmatrix} \text{ kg} \cdot \text{m}^2, \quad [\mathbf{I}_{b_2}]_{2'} = \begin{bmatrix} 0.91 & 0.18 & -0.07 \\ 0.18 & 31.90 & -0.02 \\ -0.07 & -0.02 & 31.99 \end{bmatrix} \text{ kg} \cdot \text{m}^2$$

$$[\mathbf{I}_{b_3}]_{3'} = [\mathbf{I}_{b_4}]_{4'} = \begin{bmatrix} 0.91 & -0.18 & 0.07 \\ -0.18 & 31.90 & 0.02 \\ 0.07 & 0.02 & 31.99 \end{bmatrix} \text{ kg} \cdot \text{m}^2$$

Table A.9: Inertial matrices of actuator piston assemblies.

$$[\mathbf{I}_{p_0}]_{0'} = [\mathbf{I}_{p_1}]_{1'} = \begin{bmatrix} 0.035 & 0.00 & 0.00 \\ 0.00 & 2.49 & 0.00 \\ 0.00 & 0.00 & 2.50 \end{bmatrix} \text{ kg} \cdot \text{m}^2, \quad [\mathbf{I}_{p_2}]_{2'} = [\mathbf{I}_{p_3}]_{3'} = [\mathbf{I}_{p_4}]_{4'} = \begin{bmatrix} 0.11 & 0.00 & 0.00 \\ 0.00 & 18.72 & 0.00 \\ 0.00 & 0.00 & 18.73 \end{bmatrix} \text{ kg} \cdot \text{m}^2$$

Appendix B

Parameter values used in collision-free trajectory planning algorithms

Table B.1: Parameter values of a typical rockbreaker

i	a_i (m)	b_i (m)	α_i (deg.)	$\theta_{i_{\min}}$ (deg.)	$\theta_{i_{\max}}$ (deg.)	u_i (m)	v_i (m)	ϵ_i (deg.)	$Q_{i_{\max}}$ (L/min)
1	0.37	1.10	90	-85.00	85.00	1.28	0.22	-89.04	24
2	3.06	0.00	0	-25.30	81.31	0.70	2.09	-57.63	60
3	2.38	0.00	0	-144.96	-44.37	2.07	0.73	-7.60	60
4	2.57	0.00	0	-83.95	40.19	2.12	0.63	60.56	38

Table B.2: Properties of a typical rockbreaker's hydraulic actuators

j	$\rho_{j_{\min}}$ (m)	$\rho_{j_{\max}}$ (m)	A_{blind_j} (m^2)	A_{rod_j} (m^2)
0, 1	1.07	1.5	1.03×10^{-2}	7.16×10^{-3}
2, 3, 4	1.54	2.65	2.01×10^{-2}	1.33×10^{-2}